A Quadruple-Based Text Analysis System

for History and Philosophy of Science

by

Julia Damerow

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved July 2014 by the
Graduate Supervisory Committee:

Manfred Laubichler, Co-chair
Jane Maienschein, Co-chair
Richard Creath
Karin Ellison
Wallace Hooper
Jürgen Renn

ARIZONA STATE UNIVERSITY

August 2014

ABSTRACT

Computational tools in the digital humanities often either work on the macro-scale, enabling researchers to analyze huge amounts of data, or on the micro-scale, supporting scholars in the interpretation and analysis of individual documents. The proposed research system that was developed in the context of this dissertation ("Quadriga System") works to bridge these two extremes by offering tools to support close reading and interpretation of texts, while at the same time providing a means for collaboration and data collection that could lead to analyses based on big datasets. In the field of history of science, researchers usually use unstructured data such as texts or images. To computationally analyze such data, it first has to be transformed into a machine-understandable format. The Quadriga System is based on the idea to represent texts as graphs of contextualized triples (or quadruples). Those graphs (or networks) can then be mathematically analyzed and visualized. This dissertation describes two projects that use the Quadriga System for the analysis and exploration of texts and the creation of social networks. Furthermore, a model for digital humanities education is proposed that brings together students from the humanities and computer science in order to develop user-oriented, innovative tools, methods, and infrastructures.

In Memory of

Peter Damerow (1939-2011)

ACKNOWLEDGMENTS

I thank my advisor Manfred Laubichler for his intellectual and personal support in times that were not always easy. I always enjoyed him hiding in our office from his administrative duties, pontificating about life, the universe, and everything. I thank Jane Maienschein for her input and guidance that made sure I succeeded; she always made me feel like I belong here even with my different intellectual background. My thanks go to the rest of my committee, Richard Creath, Wallace Hooper, Jürgen Renn, and Karin Ellison, who provided input and support and helped me see different perspectives of my work.

Special thanks go to Jessica Ranney for all her help and advice. She is a wonderful person and I owe her a lot. Thanks to Mikayla Madjidi who has helped me turn this dissertation into proper English and for being a wonderful friend for the last five years, making this country a home to me. Thank you also to Erick Peirson, a great friend, who listened to so much of my complaining and turned it into so many good ideas. I also thank all the students of the Digital Innovation Group; they showed me how much I love teaching and created something great out of my prototypes. Thank you to Guido Caniglia for being a great friend and all his support. Thanks to Dirk Wintergrün who put me on my path and who helped me figure out the basis of my dissertation. Thanks to all my fellow students and everybody in the Center for Biology and Society for making the last five years so much fun.

Last but not least, thank you to my family; Nikos Lessios for all the support and love that kept me going, without him I would not be where I am today; my mother and my sister who were there for me even with an ocean between us; and my father who continues to inspire me and who will always be a part of me.

TABLE OF CONTENTS

APPENDIX Page

LIST OF TABLES

LIST OF FIGURES

GLOSSARY

API	Application Programming Interface. An API specifies how other software components can interact with an application.
Eclipse Rich Client Platform	The Eclipse Rich Client Platform is a frame- work for developing desktop applications and rich clients. It provides a number of plugins that can be used to include common functionalities such as file management or text editing. Further information can be found in [The Eclipse Foundation 2014].
Java	An object-oriented, platform-independent programming language developed by Oracle.
JSF	JavaServer Faces. A Java framework for creating web applications.
LDAP Server	An LDAP server is a specific kind of server that in the case of the Quadriga-System is used for user management and authentication. For more detailed information see [Tuttle et al. 2006].
OCR	Optical Character Recognition. OCR is used to extract plain text from images or pdf files that do not have embedded text. In an image, a character is a series of dots. OCR tries to recognize characters by analyzing those dots.
SPARQL	A query language to query and manipulate RDF graphs. For more information see [W3C 2013b].
Spring Framework	The Spring Framework is an open-source Java framework that provides a developer with comprehensive functionalities such as database support, web services support, and other features. See http://projects.spring.io/spring-framework/.
Web Service	A service that can be accessed by another piece of software through a web API. For example, Conceptpower and Wordpower both provide web services for other applications to query their data.

ACRONYMS

AI	Artificial Intelligence
API	Application Programming Interface
ASU	Arizona State University
HPS	History and Philosophy of Science
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol.
JSF	JavaServer Faces
MPIWG	Max Planck Institute for the History of Science
OWL	Web Ontology Language
RDF	Resource Description Framework
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
VIAF	Virtual International Authority File
W3C	Worldwide Web Consortium
XML	eXtensible Markup Language

CHAPTER 1

INTRODUCTION

> *[…] what we are seeing is the emergence of new conjunctions between the macro and the micro, general surface trends and deep hermeneutic inquiry, the global view from above and the local view on the ground.*
> *— Lunenfeld et al. (2012b, p. 39)*

Lunenfeld et al. describe in this quote a development that they observe in the digital humanities. In contrast to close reading and careful studying of individual sources (the micro-scale), which are key methods in the humanities, *distant reading* (a term coined by Franco Moretti in [Moretti 2000][1]) in digital humanities employs computational methods to analyze large text corpora in order to find overall patterns, trends, or connections (the macro-scale) [Lunenfeld et al. 2012b]. In [Lunenfeld et al. 2012b], the authors see a "zooming in and out" between distant and close reading as a powerful tool of digital humanists. Müller calls this process "scalable reading," comparing it to the zoom function in Google Earth [Müller 2012]. He states that scalable reading enables scholars to easily switch between the details of a text and its context [Müller 2012]. Computers can support researchers by making vast amounts of data, such as texts or images, accessible through automatic extraction, analysis, and visualization of information. They can provide scholars with new tools that might help discover unknown relationships or patterns. However, they cannot replace the careful interpretation and examination of individual sources by a scholar.

In this dissertation, I will describe a research system called the "Quadriga System," which is based on the idea of representing texts as networks of concepts that can be

[1] At that point Moretti used the term "distant reading" in the context of world literature and did not focus on computational methods to automatically extract information. However, the basic idea is the same: "[d]istant reading […] allows you to focus on units that are much smaller or much larger than the text" [Moretti 2000, p. 57]

mathematically analyzed and visualized. These networks are created by scholars through close reading and structured annotation of texts. However, the Quadriga System follows a collaborative approach that facilitates the creation of a large-scale data repository in order to enable data-driven research in the History and Philosophy of Science (HPS). The system can therefore be placed in between the micro- and the macro-level of source analysis, on the so-called meso-level (or meso-scale). It is designed to help researchers detect patterns and relationships of interest in their sources by transforming the materials into structured datasets on the micro-level and analyzing them on the macro-level. The data structure underlying the Quadriga System called *Quadruples* enables scholars to seamlessly switch back and forth between a single text and a whole corpus, facilitating scalable reading.

While the Quadriga System is specifically developed for digital History and Philosophy of Science[2], I will in several places broaden the scope and discuss digital humanities in more general terms. There are many parallels between these two fields regarding methodologies as well as technological and interdisciplinary challenges and, in some regards, digital HPS can be seen as part of digital humanities. However, focusing on the history and philosophy of science, HPS is also part of the sciences and might therefore be partially positioned outside of the humanities as well. Examining the development of scientific fields and their collaboration with computer science might also inform methods in digital HPS.

The Quadriga System has been developed with the concept of the "Semantic Web" in mind, which was designed to provide structured data in a standardized way in order to

[2] I am aware that some scholars distinguish between digital and computational HPS. However, I understand the term "digital HPS" as including computational HPS and will use it throughout this dissertation.

develop more powerful applications [Berners-Lee et al. 2001]. The basic data structure of the Semantic Web is a so-called *triple*, consisting of subject, predicate, and object [Powers 2003d], for example "Julia Damerow (subject) writes (predicate) a dissertation (object)." However, for research in the history and philosophy of science, a triple typically does not provide enough information. Additional information is required to put a statement in context and specify when a statement is true. For instance, the statement above is only true in the context of this dissertation. It was not true six years ago when I wrote my Diplom thesis. Without contextual information, such a triple can therefore not be evaluated. Quadruples are designed to store such additional information by providing a reference to the source in which a given statement is made (for instance, this dissertation). Quadruples also hold information about who made a particular statement at what time and in what source.

In this dissertation, I will propose Quadruples as one solution to a question posed by Hyman and Renn in 2012: How can "human knowledge [be] adequately [represented] on the Web"? [Hyman and Renn 2012, p. 3] The Web as it exists today exhibits several shortcomings that need to be overcome to realize its full potential [Hyman and Renn 2012]. For example, Hyman and Renn criticize that collaborations are only possible to a certain extent as technologies for the creation and sharing of annotations across different types of data are lacking. Tools are needed that allow scholars to create, visualize, and analyze the relationships between documents, which include textual sources as well as other types of media [Hyman and Renn 2012]. To lift the Web to its next level, Hyman and Renn envision the Epistemic Web that will "represent not only the complete store of structured knowledge accumulated in a single lifetime by a single expert, but the collective knowledge of humanity, structured with […] care and richness" [Hyman and Renn 2012, p. 16]. In the Epistemic

Web documents can be easily annotated and annotations can be shared to combine different sources of knowledge and to detect unknown or unexpected connections [Hyman and Renn 2012]. The Quadriga System realizes some of Hyman and Renn's ideas by providing a format for annotations that is independent of the annotated media type, and by connecting these annotations creating a network of concepts that could be used to represent human knowledge as a basis for the production of new knowledge.

Another feature of the Semantic Web that the Quadriga System incorporates is a service-based architecture [Berners-Lee et al. 2001]. Several software systems interact with each other. Each one of them is independent from the rest and unaware of the particular implementations of the services with which it communicates. A service-based architecture provides flexibility and interoperability and promotes reuse of its components [Gold 2009]. The Quadriga System follows such a service-based approach in order to maximize its applicability and facilitate its maintenance. Separating functionality into different components or services ensures that any one of them can easily be replaced if required and leads to smaller modules. As a service encapsulates a particular responsibility, it can also be easily adopted and modified by projects that require similar functionality.

A large part of this dissertation is concerned with the design, implementation, and application of the Quadriga System. However, to contextualize the system, I will also discuss development, collaboration, and sharing of software tools in digital HPS. Melissa Terras states that in digital humanities "it is rare that a successful computational tool is produced which is ready to be applied beyond the projects' specific narrow research limitations" [Terras 2012b, p. 219]. Desmond Schmidt makes a similar observation when claiming that scholars in digital humanities seem often to be unable to reuse software that was developed

by other digital humanists [Schmidt 2012]. These statements are contrasted by blog posts such as [Terras 2012a] or [Gillies 2012] that announce the reuse of software (or of parts of software) that were developed for a particular digital humanities project. However, such cases seem to be the exception rather than the rule, and as Waltzer argues "[t]oo few digital humanities projects take the extra steps to argue for their generalizable value or even to create the conditions for broad adoption" [Waltzer 2012, p. 342].

Harms and Grabowski call software developed for a particular project, which focuses only on that project's needs as "specific research software" [Harms and Grabowski 2011]. "Generic research software," in contrast, is software that can be applied to various projects and provides functionality that is applicable to a broad range of research questions [Harms and Grabowski 2011]. Specific research software can become generic research software [Harms and Grabowski 2011], and I believe that this is a desired outcome for many tools developed in digital humanities. Increasing the number of tools available to a digital humanist (or digital HPS scholar) could prevent "reinventing the wheel" with every project that uses computational methods to answer certain research questions. Instead, existing software could be improved and extended by building communities of users and developers.

The question arises: How specific is software developed in the context of digital HPS? A part of this dissertation will be dedicated to analyzing the development of software and its applicability in digital HPS. I will examine to what extent digital HPS projects promote the reuse of their software, and if generic research software exists in the field. Part of this analysis is the classification of projects based on the purpose or aim of a project. The goal of a project greatly impacts what kind of software and tools are required, and how easy such tools can be generalized. For example, a project that aims to build a digital collection

will have a broader range of software available to them than a project that studies the application of a very specific text analysis method for which an algorithm might not yet exist.

Closely coupled with reuse of software is the question of the kind of software developed in the first place in digital HPS, and what collaboration is required for that development. Do multiple institutions collaborate on projects to develop software that is more generally applicable, or is software developed by one project with a very specific purpose? Also, what kind of software is typically developed for digital HPS projects? A study by Schreibman and Hanlon in 2010 describes the development of digital humanities tools and found that out of 51 tools, 54% were classified as text analysis tools, and 54% were visualization tools. In addition, many study participants developed so-called "unbranded tools" such as scripts or style sheets [Schreibman and Hanlon 2010]. Eighty-four percent indicated that their tools were publicly available for other scholars to use [Schreibman and Hanlon 2010]. However, it is not mentioned how the tools were made available. Was the source code open-source, or did the project provide an executable application for users to download?

Schreibman and Hanlon also found that 85% of the scholars involved in the development of tools collaborated with programmers [Schreibman and Hanlon 2010]. Collaboration is a typically characteristic of digital humanities projects, as they often require expertise from multiple disciplines (for example history and computer science) [Lunenfeld et al. 2012a]. Along with collaboration, however, come difficulties such as communication barriers. Humanists and computer scientists especially often lack a common language [Terras 2012b; Siemens et al. 2009]. The Digital Innovation Group at Arizona State University prepares students from the Biology and Society program and the computer science

department for these kinds of situations by engaging them in digital HPS projects that involve software development and research tasks. I will describe the structure and mission of that group as a possible model for similar endeavors.

This dissertation is structured as follows: In the next chapter (The Present State), I will first give a general introduction to digital humanities and its history. I will then describe the present project landscape of digital HPS. I will evaluate what kind of projects exist and identify a gap that, to my knowledge, exists in the field. The Semantic Web plays an important role for the Quadriga System, and I will therefore provide a brief overview of this concept. Finally, I will contrast the field of digital HPS with the fields of medical informatics and bioinformatics, which profited greatly from incorporating computer science and could inform similar developments in digital humanities/HPS.

Chapter 3 will provide a brief overview of technologies relevant to this dissertation. I will then describe in Chapter 4 the core of this dissertation, the Quadriga System. The chapter will detail fundamental concepts such as Quadruples and how they form networks. It will also illustrate how such Quadruple networks can be used and visualized. I will describe the architecture of the Quadriga System before giving a brief summary of the implementation of each component of the system.

In the subsequent chapter, I will present two projects that use the Quadriga System: the EP Annotation Project and the Genecology Project. Both projects use the software of the Quadriga System to generate Quadruple networks. However, the focus of each project is different. While the EP Annotation Project aims to develop new ways to explore their data (the Embryo Project articles), the Genecology Project creates collaboration networks for mathematical analysis and visualization. Chapter 6 describes the Digital Innovation Group

that contributed greatly to the development of the software that I prototyped for my dissertation. The group is an example of how students could be trained for work at the intersection of computer science and history and philosophy of science while contributing to research projects. In Chapter 7: Future Work, I will outline possible future developments for the Quadriga System. Mostly these are concrete, straightforward extensions to existing parts of the system. Other development suggestions, however, are aimed at the bigger context of the system. The last chapter (Chapter 8) provides a brief conclusion of my dissertation and will summarize the results. The appendix that follows contains supplementary data that I will refer to throughout the chapters.

CHAPTER 2

THE PRESENT STATE

The research system that I describe in this dissertation has been developed in the context of digital History and Philosophy of Science. Digital HPS, as part of the field of history and philosophy of science, combines digital and computational methods with traditional scholarly methods in history and philosophy of science research [Maienschein and Laubichler 2012]. As I will show in section 2.2, while a big part of all projects in digital HPS are being developed to support traditional research methods such as close reading and examination of sources, some projects provide new methods that use automatic analysis or big data approaches. The research system I am proposing tries to bridge these two extremes by offering tools to support close reading and interpretation of texts, while at the same time provide a way for collaboration and data collection that could lead to analyses based on big datasets.

I believe that digital HPS as a field is still in its beginnings and that looking at other fields might provide valuable insights for its future development. Concepts developed in the context of computer science, such as the Semantic Web, could on the one hand provide useful applications to digital HPS, and on the other hand might benefit from insights that the history and philosophy of science has to offer. Fields that emerged from an incorporation of computer science techniques, such as the field of bioinformatics, could inform digital HPS scholars about possible perspectives and approaches for their field.

In section 2.1 I will describe the development of the field of digital humanities, as digital HPS has many associations with this field and several overlapping research interests. I will then examine the present state of digital HPS in section 2.2. Although a part of digital

humanities, digital HPS is a distinct field with a different focus and history. In section 2.3, I will give some background information on the semantic web and how it relates to digital humanities. The last section (section 2.4) is focused on bioinformatics and medical informatics and how these fields developed. I will especially analyze how their development can inform future progress in the field of digital HPS.

2.1 Digital Humanities

In the foreword to "A Companion to Digital Humanities," Father Roberto Busa writes: "[d]uring World War II, between 1941 and 1946, I began to look for machines for the automation of the linguistic analysis of written texts. I found them, in 1949, at IBM in New York City." [Busa 2004, p. xvi] That year Busa, who is considered by many as "the pioneer of the field of humanities computing" [Svensson 2009, §17], started his work on the *Index Thomisticus*, an index of all the terms and their contexts in the works of Thomas Aquinas. Busa began by using punch cards, later switched to magnetic tapes, and eventually to CD-ROMs and hard drives. Today the *Index Thomisticus* is over 1GB in size and can be queried through a web interface[3]. [Busa 2004; ADHOb; Svensson 2009]

With his work, Busa was one of the first in a field called "humanities computing." Other researchers soon started to work on similar projects and to research how computers could aid in the humanities, especially with regard to the analysis and exploration of texts [Hockey 2004]. In 1966, the first journal of the field, "Computers and the Humanities," was published. This was followed by the founding of the Association for Literary and Linguistic Computing (ALLC) in 1973, and the Association for Computers and the Humanities (ACH)

[3] See http://www.corpusthomisticum.org/it/index.age

in 1978 [Lunenfeld et al. 2012a]. ALLC's original purpose was to support "the application of computing in the study of language and literature" [EADH, § Our History]. However, with the growing of the field the organization broadened its scope to include other areas such as history, music, or image processing [EADH]. To reflect that change, ALLC was renamed in 2012 to the European Association for Digital Humanities (EADH) [EADH].

As the field of humanities computing developed, the understanding of its role in respect to the "traditional" humanities changed as well. According to Berry, during "the early days [humanities computing was] often seen as a technical support to the work of the 'real' humanities scholars" [Berry 2011, p. 2]. However, he continues that "as the projects became bigger and more complex, and as it developed computational techniques as an intrinsic part of the research process, technically proficient researchers increasingly saw the computational as part and parcel of what it meant to do research in the humanities itself" [Berry 2011, p. 2]. To demonstrate this change in how researchers in the field understood their work, the term "digital humanities" emerged. An indication that this term would permanently replace the previous used name "humanities computing" was the publishing of Blackwell's *Companion to Digital Humanities* in 2004. Some authors even claim that the *Companion* initiated (see [Hayles 2012]) or at least finalized that name change and that since then "the name has stuck" [Fitzpatrick 2012, p. 13].

By changing the name of the field to "digital humanities," scholars in that field did not only want to signal that their understanding of the role of their work changed. In the beginning most projects, similar to Busa's *Index Thomisticus*, were focused on working with texts and literature using computational methods [Berry 2011]. While this area still plays an important role in digital humanities, the scope of the field has broadened and now includes

other areas with different kinds of source material such as musicology or media studies [Fitzpatrick 2012]. When discussing a title for the *Companion to Digital Humanities*, Blackwell, for that reason, argued against "Companion to Humanities Computing" [Schreibman 2012; Fitzpatrick 2012].

In 2002, efforts began to create an "umbrella organization" for the different organizations in the field of digital humanities such as the ALLC (or EADH), ACH, or the Australasian Association for Digital Humanities (aaDH) (see [ADHOa] for a full list). The organization was named the Alliance of Digital Humanities Organizations (ADHO). ADHO organizes a yearly conference, the "Digital Humanities Conference," which was first held in 1989 at the University of Toronto, with its 25-year anniversary in 2013 at the University of Nebraska-Lincoln, USA. [ADHOa]

Berry, as well as Hayles, both use the *Digital Humanities Manifesto 2.0*'s[4] classification that groups digital humanities projects into waves [Berry 2011, Hayles 2012]. According to Berry, the first wave "involved the building of infrastructure" [Berry 2011, p. 3]. An example of this is the building of digital repositories and marking up texts for use in computational processes. As digital humanities (or humanities computing) has its roots in computational linguistics, linguistic analysis like term frequency analysis also played an important role during that initial phase of digital humanities [Lunenfeld et al. 2012a]. The second wave, however, broadened the scope of digital humanities and the existing infrastructure to include other digital materials (besides texts) and what Presner (cited by Berry) calls "born-digital"

[4] By Schnapp and Presner, see http://jeffreyschnapp.com/wp-content/uploads/2011/10/Manifesto_V2.pdf

materials [Presner 2014, Berry 2011], for example, electronic texts that were not digitized but created for electronic publication only, websites, or web applications.

In addition to the first and second wave of digital humanities projects, Berry suggests a possible third wave centered on the digital and computational parts of digital humanities projects. He puts forward the idea that digital humanities projects of the third wave might be concerned with "the way in which digital technology highlights the anomalies generated in a humanities research project and which leads to the questioning of the assumptions implicit in such research" [Berry 2011, p. 4]. Hayles proposes a similar idea when discussing how scale might change humanities research [Hayles 2012]. She refers to Gregory Crane, who states that computers enable scholars to get an overview of the continuously growing corpus of relevant literature that, due to its size, they would not be able to comprehend otherwise. Hayles also quotes Franco Moretti, who proposes that "distant reading" (analyzing texts by using only the results of computational analysis) as a method for literary history. While I do not believe that distant reading could or should replace traditional "close reading" approaches[5], I do agree that new technologies in digital humanities can inform the work of researchers by presenting new methods and approaches. Throughout my dissertation, I will come back to that topic and how the research system I describe relates to it.

2.2 The Digital HPS Landscape

Digital History and Philosophy of Science, although steadily growing, is still a very young field. Some digital HPS projects date back to 2002 (for example, the Archimedes Project [Harvard University 2004]) or even earlier. However, it was not until 2011 that the Digital

[5] The results that the careful analysis and interpretation of individual texts by a human reader generates, can not be reproduces by techniques that are designed to detect large-scale patterns in data.

HPS Consortium [Digital HPS Consortium 2013a] was formally founded with the goal to "develop, support, and promote digital HPS projects, including editing, publishing, and scholarly tools" [Digital HPS Consortium 2013a][6].

In this section, I will give an overview of what kinds of digital HPS projects exists. This overview will put the Quadriga System into the context of current efforts in the field. I will point out the gap that I have identified in the field and how the research system I am proposing could be a step towards closing that gap.

To get an overview of the landscape of digital HPS projects, I reviewed all the projects listed on the website of the Digital HPS Consortium [Digital HPS Consortium 2013b]. In addition, I went to the websites of all the institutions listed as participating institutions and reviewed all the digital HPS projects (or projects closely related to digital HPS) that were listed there as well. Although not complete, the website of the Digital HPS Consortium is the only source that provides a catalog of digital HPS projects. Using the website, I identified a total of 43 projects. I classified these projects using the categories and criteria described in Table 1 (see Appendix A for details on the classification of individual projects).

Table 1: Digital HPS Project Categories

CATEGORY	DESCRIPTION
Digital Collection	Any project with the main purpose of collecting, digitizing, and providing the resulting material online.
Digital Collection w/ computational tools	Any Digital Collection project that in addition to making sources available on the internet provides analysis functionality for the provided material, or develops

[6] According to the Digital HPS website, the earliest meeting of scholars in the field of digital HPS was in April 2009. However, the Digital HPS Consortium was not officially founded until 2011.

CATEGORY	DESCRIPTION
	computational tools for creating the collection in the first place.
Digital Collection w/ data collection	Any Digital Collection project that creates a digital collection with the purpose of collecting data for further research about the provided material.
Digital Collection/ Education	Any Digital Collection project that uses the collection creation process for educational purposes in form of classes, online tutorial, etc.
Online Bibliography	Any bibliography that provides its content via a searchable web interface.
Online Repository	Any repository that stores electronic material and makes it accessible through a web interface. In contrast to digital collections that have the goal to present carefully selected materials (often combined with additional information), the focus of online repositories is the storage (and not presentation) of documents and their metadata.
Website	Any project with the main purpose of creating a website to distribute information, which does not fall into any of the above categories.
Virtual Exhibition	Any project that classify themselves as "virtual exhibition," or "virtual exhibit," or any other project that virtually represents a physical space and provides digital material distributed in that "virtual space."
Project using Computational Tools	Any project that provides researchers with computational tools for the analysis of sources.
Computational Tools Development Project	Any project that is concerned with developing software tools.

Out of the forty-three reviewed projects, seven projects were computational tools development projects. Those projects differ from the projects of other categories, as they are

not directly concerned with the presentation, creation, or dissemination of specific research data[7]. Therefore, I will exclude the seven projects from the following statistics and will come back to them later on.

Figure 1 shows how many projects fall into each of the nine categories (excluding computational tools development projects). Out of 36 projects, 19 have been classified as "pure" digital collections. All projects that were classified as digital collection, digital collection with computational tools, digital collection with data collection, or digital collection/education even make up 25 of the 36 digital HPS projects. Online bibliographies, online repositories, virtual exhibitions, and website projects together make up eight of the

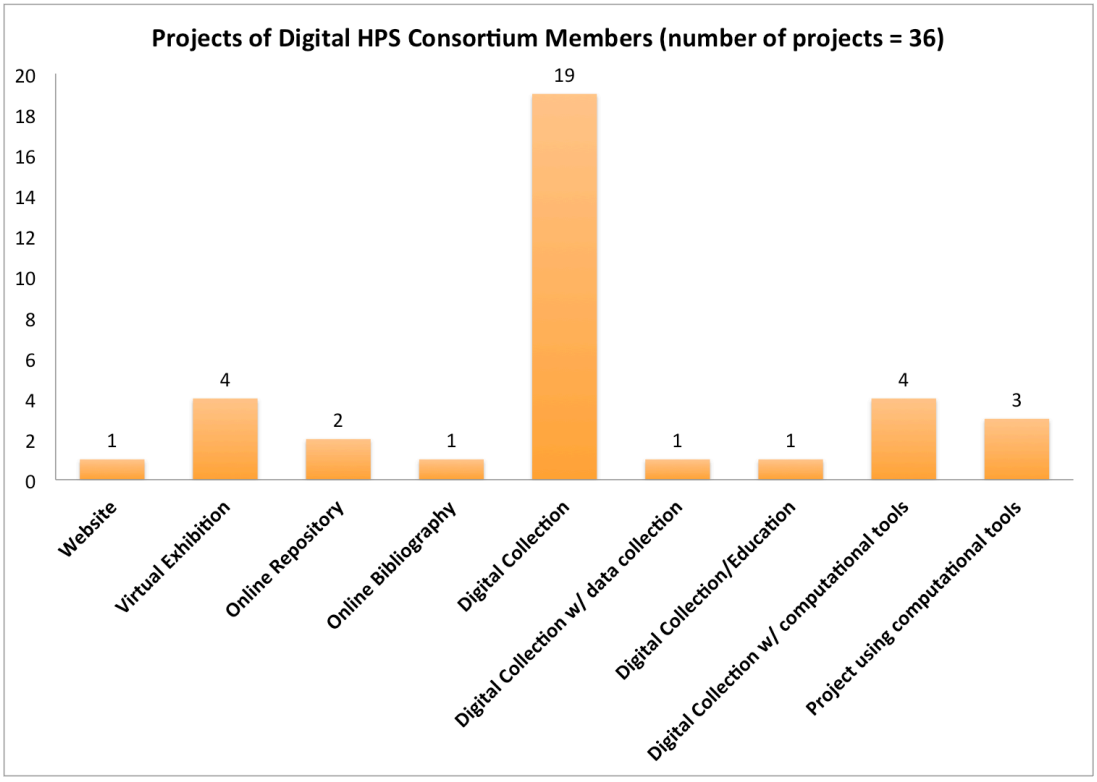

Figure 1: Projects of Digital HPS Consortium Members

[7] I use the term "research data" in a very broad sense. It could refer to documents, full texts, videos or any other type of data used by historians and philosophers of science.

projects. Only three of all digital HPS projects have been classified as projects using computational tools.

2.2.1 Mai/Lau Types and Use of Computational Tools

In [Maienschein and Laubichler 2012], Maienschein and Laubichler describe three types of projects (in the following referred to as Mai/Lau types). First, there are projects that make "published and archived objects of traditional scholarship available in digital form" [Maienschein and Laubichler 2012, p. 39]. I will call such projects "traditional" projects. Second, they list scholarly works that were created specifically for digital publication ("digital" projects). And third, Maienschein and Laubichler state that digital HPS researchers can come to new conclusions "that result from use of new tools […]" [Maienschein and Laubichler 2012, p. 39]. I will call these "computational" projects. Using the Mai/Lau types, the categories I used to classify digital HPS projects can be assigned types as shown in Table 2.

Table 2: Digital HPS Project Categories and Mai/Lau Types

CATEGORY	MAI/LAU TYPE
Digital Collection	Traditional projects
Digital Collection w/ computational tools	Digital projects/computational projects
Digital Collection w/ data collection	Digital projects
Digital Collection/Education	Digital projects
Online Bibliography	Traditional projects
Online Repository	n/a
Website	Digital projects

CATEGORY	MAI/LAU TYPE
Virtual Exhibition	Digital projects
Project using Computational Tools	Computational projects

Traditional and digital projects are similar in nature. Both types of projects are mainly concerned with making sources or information digitally available. They use computational tools[8] for *source presentation and/or dissemination*. Computational projects, however, have a different goal. Those projects employ computational methods and computational tools as part of their research methods in order to create or analyze research data (*data creation and/or analysis*). Projects in the category "digital collection with computational tools" are hybrids between digital and computational projects. They use computational methods to provide additional analysis functionality of sources, on top of making those sources available online. Or, they create collections that are outcomes of using computational methods and tools. Table 3 shows the categories I used to classify digital HPS projects and how they use computational tools.

Table 3: Digital HPS Project Categories and Use of Computational Tools

CATEGORY	USE OF COMPUTATIONAL TOOLS
Digital Collection	Source presentation and/or dissemination
Digital Collection w/ computational tools	Source presentation and/or dissemination and data creation and/or analysis
Digital Collection w/ data collection	Source presentation and/or dissemination
Digital Collection/Education	Source presentation and/or dissemination

[8] I use the term "computational tools" in a very broad sense, referring to any kind of software tool.

CATEGORY	USE OF COMPUTATIONAL TOOLS
Online Bibliography	Source presentation and/or dissemination
Online Repository	Source presentation and/or dissemination
Website	Source presentation and/or dissemination
Virtual Exhibition	Source presentation and/or dissemination
Project using Computational Tools	Data creation and/or analysis

2.2.2 Data Creation/Analysis vs. Presentation/Dissemination

Figure 2 shows how many of all reviewed digital HPS projects[9] are computational projects and how many are concerned with source presentation and/or dissemination. While there

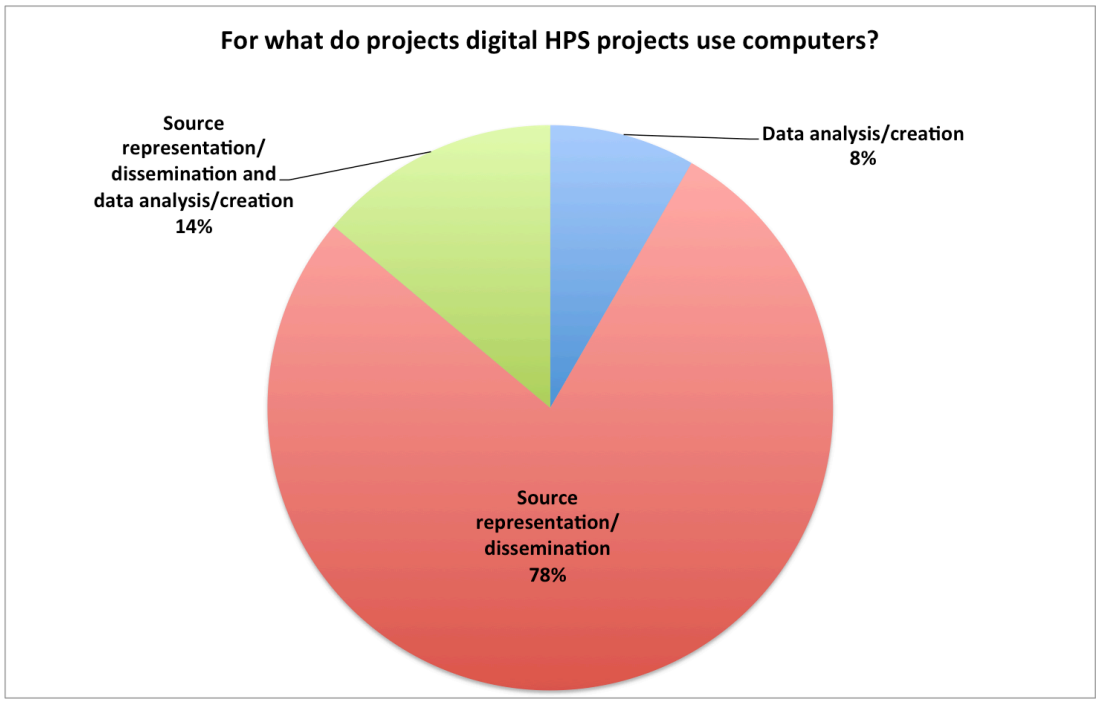

Figure 2: Digital HPS Projects by Main Objective

[9] At this point, I am still excluding computational tools development projects.

are 78% data presentation/dissemination projects, only 8% of the listed projects are projects that use computational methods to create or analyze data.

The imbalance between using computers to present and disseminate sources and using computers to aid the analysis or creation of data might be explained by looking at the methods of the field of history and philosophy of science. Traditionally, researchers in the humanities search archives and libraries for relevant sources, study and analyze those sources, make notes and annotations, and eventually publish their findings [Bordoni 2007; Lunenfeld et al. 2012a]. With the turn of the digital age and the accompanying increase of computing power, it was a logical next step to facilitate access to source material by creating online collections and archives. However, as Maienschein and Laubichler state, computers can be used for more than publishing and providing sources online. By employing computational methods, researchers might be able to come to new or different conclusions. For example, computers could help them finding "relationships among items not known to be linked" [Maienschein and Laubichler 2012, p. 39], or analyzing huge amounts of data that would not be possible to analyze without computing power.

Explaining the imbalance seen in Figure 2 only by looking at the methods of the field, however, might not be satisfactory. The Archimedes Project, for example, has existed for over ten years now and uses computational tools to enhance the digital collection it provides. A question arises about why this kind of digital collection isn't more common, as the idea of using computational tools to do more than "just" source presentation and dissemination has been around for over ten years. To answer that question, I collected information about the collaborators on the digital HPS projects I classified (see Figure 3). I used the Digital HPS Consortium website as a starting point and then looked at the project

websites (if they existed) to find information about the participating institutions on the different projects. For the following analysis, I also included projects that are concerned with developing computational tools, as there might be a connection between the development of computational tools and their application in projects.

Figure 3 shows that 19 out of 43 projects (≈ 44%) have only one participating institution. Thirteen projects (≈ 30%) are run by 2 institutions. Eight projects have between three and five participating institutions, and only three projects have more than five institutions collaborating. More than half of all digital collection projects (17 out of 29 ≈ 59%) have two or more participating institutions. In contrast, projects that use computational tools do not have collaborators at all. Projects that are concerned with the development of computational tools, however, have between one and three participating institutions.

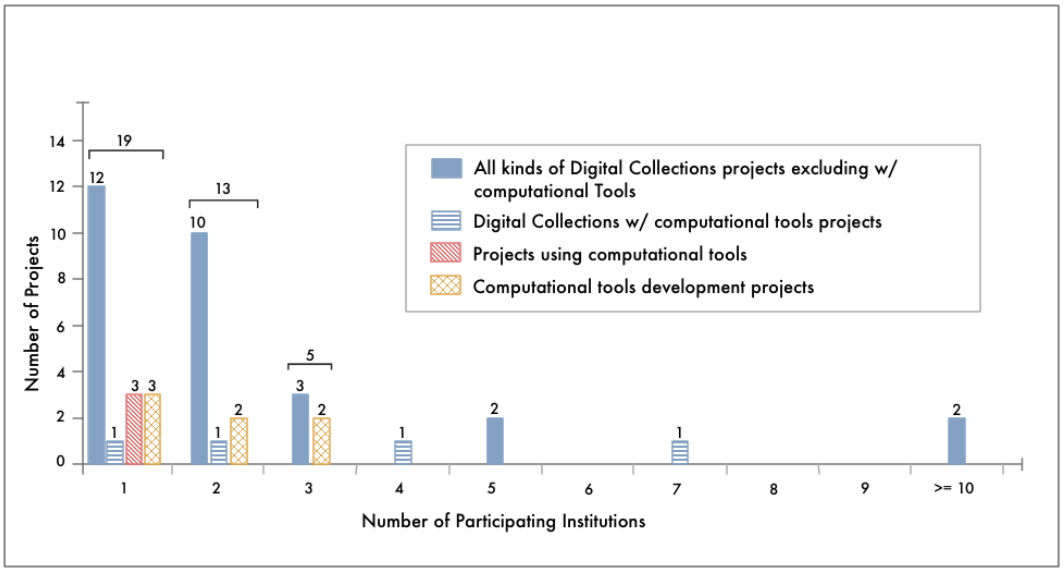

Figure 3: Number of Collaborators on Digital HPS Projects

This discrepancy between the number of collaborators on projects using or developing computational tools and digital collection projects might provide an explanation for the imbalance between these types of projects. More collaborators might result in better funding situations and more resources (materials as well as expertise). Also, collaborating with an institution that already has experience with a specific type of project might make a project more likely to succeed. Looking at the institutional landscape of digital HPS projects might provide some further insights. Figure 4 shows all the institutions that are participating in more than two digital HPS projects.

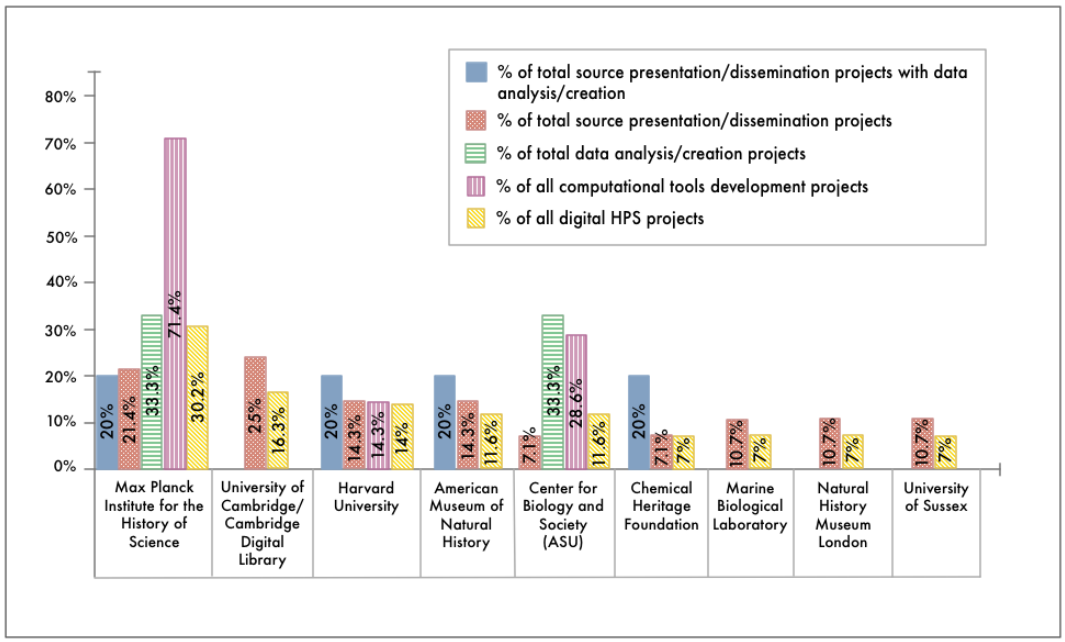

Figure 4: % of Projects per Category for Institutions with more than two Projects

Figure 4 shows that of the nine institutions, only three institutions are participating in developing computational tools: the Max Planck Institute for the History of Science, Harvard University, and the Center for Biology and Society at Arizona State University (ASU). Regarding projects concerned with data creation and analysis, only the Max Planck

Institute for the History of Science (MPIWG) and the Center for Biology and Society work on such projects.

Given that there is no collaboration on projects using computational tools, and a single institution participates in over 70% of all tools development projects, this means that the MPIWG plays a key role regarding computational tools, followed by the Center for Biology and Society. This emphasis on two institutions could influence the "shape" of the digital HPS tools landscape in the way that it is probable that these institutions will develop tools according to their research foci. Such tools might not be reusable by other projects with different project goals, resulting in so few projects using computational tools.

For a detailed analysis of how well software developed in the context of digital HPS projects can be reused, Table 4 gives an overview of the reviewed tools development projects and the uses of those tools. Table 5 provides an overview of all projects using computational tools.

Table 4: Computational Tools Development Projects

PROJECT
Agnostic Editor *URL*: http://etcetera.caret.cam.ac.uk/blog/agnostic-editor *Description*: The Agnostic Editor is being developed to simplify the editing process of structured data. It is optimized to work with XML texts. Its goal is eliminate the need to train editors of structured data such as XML text in "complex theory of structures." (Martin 2013) *Used for*: Data editing as part of the preparation process of data for presentation and dissemination. *Available as desktop/web application*: no *Source code available*: no *Documentation available*: no
Anteater

PROJECT
URL: http://anteater-tool.sourceforge.net/ *Description*: Anteater is a text mining web application to extract certain information from Federal Register documents regarding endangered species research. *Used for*: Extraction of information from text documents. *Available as desktop/web application*: yes (web) *Source code available*: yes *Documentation available*: yes
Arboreal *URL*: http://arboreal.sourceforge.net/ *Description*: "Arboreal MWN is a tool for content-based access to XML documents." (Damerow 2013) It provides functionality to read and edit XML documents, and to analyze those by using a language analysis service. *Used for*: XML editing as for example part of the preparation process of presentation and dissemination of XML texts. *Available as desktop/web application*: yes (desktop) *Source code available*: yes *Documentation available*: yes
Digilib *URL*: http://digilib.sourceforge.net/ *Description*: Digilib is a web application to view images in a web browser. "[D]igilib enables very detailed work on an image as required by scholars with elaborate viewing features like an option to show images on the screen in their original size." (Casties and Raspe 2013) *Used for*: Presentation and dissemination of image data. *Available as desktop/web application*: yes (web) *Source code available*: yes *Documentation available*: yes for developer
Digital Scrapbook *URL*: http://www.mpiwg-berlin.mpg.de/en/research/projects/DEPT1_10_30Buettner-DigitalScrapbook *Description*: "The digital scrapbook supports the full spectrum of source-based scholarly

PROJECT
work, from the first annotation to the final publication, in a unified format supported by the same tools." (Renn et al.) The Digital Scrapbook lets researchers share their collections, annotations, and references by making them accessible through the web. *Used for*: Collecting and sharing of data such as documents, annotations, or references. *Available as desktop/web application*: no *Source code available*: no *Documentation available*: no
Omeka *URL*: http://omeka.org/ *Description*: Omeka is a web-publishing platform to publish collections and virtual exhibitions of for example libraries or archives. "Omeka falls at a crossroads of Web Content Management, Collections Management, and Archival Digital Collections Systems." (Roy Rosenzweig Center for History and New Media 2013) *Used for*: Presentation and dissemination of documents. *Available as desktop/web application*: yes (web) *Source code available*: yes *Documentation available*: yes
Virtual Spaces MWN *URL*: http://virtualspaces.sourceforge.net/ *Description*: Virtual Spaces MWN is a tool to create virtual tours containing images, texts, and videos. These virtual tours can be exported and presented as webpages. (Damerow 2009) *Used for*: Presentation and dissemination of texts, images, and videos. *Available as desktop/web application*: yes (desktop) *Source code available*: yes *Documentation available*: yes, but incomplete

Table 5: Projects using Computational Tools

PROJECT
Archimedes Project *URL*: http://archimedes.fas.harvard.edu/

PROJECT

Description: The Archimedes Project developed an "interactive environments for scholarly research on the history of mechanics and engineering from antiquity to the Renaissance" (Harvard University 2004).

Available as separate desktop/web application: partially (both)

Source code available: partially

Documentation available: partially

Art of Life

URL: http://biodivlib.wikispaces.com/Art+of+Life

Description: The Art of Life project aims to develop software to automatically identify and describe images in the Biodiversity Heritage Library.

Available as separate desktop/web application: yes (web)

Source code available: yes

Documentation available: partially

Chymistry Of Isaac Newton

URL: http://www.chymistry.org

Description: The Chymistry Of Isaac Newton project provides an online edition of Isaac Newton's alchemical manuscripts. It also provides a number of online tools for these texts such as a glossary or a tool analyze the relationships between terms, text parts, and documents in the Newton corpus.

Available as separate desktop/web application: no

Source code available: no

Documentation available: yes, for provided online tools

Cultures of Knowledge

URL: http://www.culturesofknowledge.org/

Description: This project provides an online catalogue of letters sent in the 16th, 17th, and 18th century. In addition to the catalogue, the project aims to develop computational tools to analyze the data in the catalogue.

Available as separate desktop/web application: no

Source code available: no

Documentation available: no

PROJECT
Indiana Philosophy Ontology Project (InPho) *URL*: https://inpho.cogs.indiana.edu/ *Description*: This project uses the articles from several sources (such as the Stanford Encyclopedia of Philosophy) to create a "dynamic ontology" using data mining techniques. *Available as separate desktop/web application*: no *Source code available*: yes *Documentation available*: yes, in source code
Using "Gene Knock-Out" Techniques to Test Cultural Evolution *URL*: http://devo-evo.lab.asu.edu/cultural-evolution *Description*: This project employs topic modeling techniques to answer the question if there exist "functional units" in historic literature collections that significantly influence the vocabulary and its structure of the corpus. *Available as separate desktop/web application*: no *Source code available*: no *Documentation available*: no
Science under Scrutiny *URL*: http://www.mpiwg-berlin.mpg.de/en/news/features/feature28 *Description*: This project tries to understand how policies regarding research (especially biological field research) are shaped by social norms and values. As part of this project, a text extraction tools was developed mining federal regulatory documents for information on endangered species research. *Available as separate desktop/web application*: yes (web) *Source code available*: yes *Documentation available*: partially, in source code

Table 4 shows that five out of seven tools can be used for presentation and/or dissemination of data such as documents, texts, or images, or during the preparation process of such data. This means that they are very likely used in digital collection projects. The three projects that use computational tools, however, are concerned with:

- Text mining (see the project "Science under Scrutiny");
- Machine reasoning and data mining (see the InPhO project for example (Buckner et al. 2010]);
- Topic modeling (see the project "Using 'Gene Knock-Out' Techniques to Test Cultural Evolution").

Similarly, projects with the goal to build digital collections in combination with computational tools, use their tools for:

- Morphological analysis (Archimedes project);
- Image analysis (Art of Life project);
- Correlation between documents and terms (Chymistry Of Isaac Newton);
- Analysis and visualization of metadata (Cultures of Knowledge).

The web application Anteater was developed for Etienne Benson's project "Science under Scrutiny." Besides that, there seems to be no overlap between tools and projects using computational tools. The question arises: How are computational tools developed in the context of digital HPS being used? Are they used, but not mentioned? Or, were they created for projects that cannot be found through the website of the Digital HPS Consortium?

Eight out of thirteen projects (projects that use computational tools or tool development projects) have made the code of their software open-source. Six of these eight projects have a web or desktop application that can be downloaded by a user. Out of the six projects that use computational tools, three make their software at least partially available as separate software applications. However, two of them (Anteater/Science under Scrutiny and Art of Life) are very specific to the project they were developed for and would probably require a programmer to adapt them to new projects. They are what Harms and Grabowski

call "specific research software" [Harms and Grabowski 2011]. Overall, it seems to be the case that projects using computational tools tend to develop those tools tightly coupled with the projects themselves. And, although they make their source code available, they do not promote their tools as standalone products to be reused by other projects.

Another important factor that should be considered in the development of a tool is documentation and user interface. In [Gibbs and Owens 2012], the authors state that in the context of digital humanities "[m]any tools now seem to downplay the importance of the user interface and documentation" [Gibbs and Owens 2012, §33]. It is often assumed that a "really interested" user will teach himself how to use a tool and how to use it on his own data [Gibbs and Owens 2012]. By making this assumption when developing software for a project, it becomes much less likely that other projects will be able to adopt the software. A similar trend can be observed in the above described tools and projects. In many of them, documentation is not or only partially available, or the software is only available in the form of libraries or scripts requiring a programmer or at least "computer-savvy" person.

There are two characteristics of digital HPS projects that might explain the imbalance of numbers of digital collection projects and projects using computational tools. First, projects that use computational tools tend to develop computational tools as part of the project without promoting these tools separately. Second, projects that are concerned with the development of software tools are mainly developing tools for presentation and dissemination of data. There is already a lot of software developed to create digital collections. However, if a project's goal is to use computational tools to analyze or create data, these tools in many cases need to be specifically developed for the project. Even if

there are other projects with similar objectives, the computational tools that are used in these other projects might simply not be available.

2.2.3 Conclusion

In summary, I believe that the current digital HPS landscape can be characterized as follows. First, most digital HPS projects are concerned with the presentation and dissemination of data. Second, for the few projects that use computational tools there is very little collaboration. Third, most software tools that are being developed by the digital HPS community are indented to be used for the presentation and dissemination of data. And fourth, computational tools that are being developed for the analysis and creation of data are not being promoted for reuse by other projects.

The statements above are all based on my initial review of digital HPS projects. For this review I used the digital HPS Consortium website as starting point and tried to find as much information as possible about digital HPS projects. It is very likely that there are more projects that simply cannot be found through the digital HPS Consortium website. Moreover, many of the project websites lack critical information such as tools being used, or project goals. In addition, the computational tools being used are often not made available or are not sufficiently documented for reuse.

For the Digital Humanities Conference 2010, Melissa Terras identified this problem as being widespread in the digital humanities. She states, "we all should be taking our digital identity and digital presence a lot more seriously." [Terras 2010, Section 4. Digital Identity] Better documentation and promotion of digital/computational tools might increase reuse and adaption of existing tools and might lead to a more balanced digital HPS project landscape.

2.3 Semantic Web

> *The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation.*
>
> — [Berners-Lee et al. [2001, p. 37]

In 2001, Tim Berners-Lee et al. proposed in [Berners-Lee et al. 2001] the idea of the Semantic Web. They envisioned the Semantic Web to extend the "traditional" web with the goal to enable machines to process data automatically.

In the traditional web, webpages are formatted using Hypertext Markup Language (HTML). Using HTML, text can be structured into elements such as headings, subheadings, or paragraphs, and characters can be formatted, for instance, made bold. However, semantic markup is only possible to a minimal extent. For example, expressing that a certain name refers to a person or represents a date is not easily possible.[10] Hyperlinks are used to link between webpages. However, such hyperlinks do not specify the type of a link. For example, if one webpage contains the biography of a scholar and links to his or her publications, a hyperlink cannot express that there exists an authorship relation between the person and a publication.

Tim Berners-Lee et al. envisioned that in the Semantic Web, pieces of software often called "software agents" or "agents" would be able to carry out tasks such as making appointments, or finding the right store for purchasing certain items. They would do this automatically analyzing and using data available from different sources on the web. In the traditional web, most webpages contain unstructured data. This poses a problem for a

[10] This has changed to a certain extent with the release of HTML5 in 2008 (see http://www.w3.org/TR/html5/).

software application that tries to use the data. For instance, for a human it is simple to find the opening hours of a store on a webpage. However, for a computer this is a complicated task. Not every store website uses the same format to display their opening hours; some might label it "opening hours," some might call it "store hours," or some might simply say "we are open." Moreover, the webpage might say "closed on holidays," which would require the agent to understand that this phrase means the store is not open, and to know in which state of the US a store is located, as well as the relevant holidays of that state. In the Semantic Web, such information could be encoded and published in addition to the traditional website in a format that makes it easy for software agents to "understand" and use the information. [Horrocks 2008]

A concept that is tightly coupled with the idea of the Semantic Web is the concept of ontologies. This notion of ontologies is different from ontologies in philosophy. "An ontology [in computer science] is a shared and common understanding of some domain that can be communicated across people and computers" [Benjamins et al. 2004, p. 434]. Such ontology usually describes a domain by describing the "concepts" in that domain, and how they relate to each other. A common example is an ontology of pizza toppings. It describes what kind of pizza toppings there are, and how many toppings a pizza can have, etc. An ontology is often created by a group of experts in the domain that the ontology describes. I will discuss the concept of ontologies in more detail in section 3.2.

Ontologies are used in several areas, such as in bioinformatics or artificial intelligence [Ceusters and Smith 2011]. For example, [Berkeley Bioinformatics Open Source Project 2013] is a collection of ontologies used for domains such as anatomy, proteins, or experiments. "The humanities are a particularly difficult area for the development and

application of ontologies." [Burrows 2011, p. 187] One of the reasons for this is that humanities research is often concerned with long periods of time, in which the meaning of terms and their context can change. Moreover, the context of a term might be ambiguous or the interpretation of a term might depend on its context. Another difficulty is that humanities researchers oftentimes work with several "worldviews and ways of categorizing the world" [Burrows 2011, p. 187], which poses a problem for working with ontologies that are created with the goal to describe a common understanding. [Burrows 2011]

Burrows lists two important ontologies in the humanities: VICODI[11] and CIDOC-CRM[12]. The VICODI (Visual Contextualization of Digital Content) project aimed "to enhance people's comprehension of digital content on the Internet" [Nagypal et al. 2005, p. 328]. Its goal was to develop a Semantic Web application that uses an ontology of European history, and which would provide functionality to explore historical documents beyond full-text search. One of the main insights of the VICODI project was that the context of a document would be an important factor in what Nagypal et al. call the "user context." This user context could be used to create better search results over the document corpus. Nagypal et al. give the example of a user who reads a document about World War I. The user's context would in this case contain World War I. If this user subsequently searches for information about Serbia, it would be likely that documents about Serbia from the beginning of the 20th century would be more interesting to the user than documents about the Kosovo conflict in 1998. [Nagypal et al. 2005]

---

[11] See [Information Society Technologies 2004] for the project website.
[12] See [Doerr 2013] for the project website

Part of the VICODI project was the development of an ontology about European history. The fundamental problem the project had to solve was to build an ontology that captured all the necessary information "to be 'proper' from a historical point of view" [Nagypal et al. 2005, p. 336] while at the same time building an ontology with a simple enough structure to be used by computer scientists to develop algorithms that would work on user and document contexts. To solve this problem, Nagypal et al. developed a rather flat ontology (a flat hierarchy of elements) that was detailed enough to satisfy historians, but which was also simple enough for computer scientists. However, besides this problem, which is not exclusive to the study of history, Nagypal et al. identify four additional history related complications. [Nagypal et al. 2005]

TIME DEPENDENCE In a field such as history (or history of science), almost everything that is of interest has a temporal component. For example, the borders of countries change over time, names of places change over time, or the number of children a person has changes over time.

UNCERTAINTY Historians often research questions that arise from missing documents or contradicting sources. Nagypal et al. give as example the birth of Stalin, which is still debated among historians. Officially, in the Soviet Union his birth is registered as being 21 December 1879. However, church records say that Stalin was born 6 December 1878.

SUBJECTIVITY In the study of history, events[13] are often subject to interpretation. Concepts such as certain periods in time are often not clearly defined and

[13] I am using the term "event" in a very broad sense here. An event would be World War II, the birth of Martin Luther King, or the women's rights movement in Europe.

different historians have their own subjective interpretations of them. Nagypal et al. give as examples periods such as "Enlightenment" or the "Middle Ages" for which start and end dates are not agreed upon.

WHY QUESTIONS Ontologies represent "facts" and are useful to answer questions such as where and when did an event happen, what happened, and who was involved. However, for historians such questions are not as interesting as why an event happened. For example, an historian could ask why the Red Army was so frequently defeated in 1941. According to Nagypal et al., historians are interested in the context of facts and not just the facts.

Nagypal et al. solved the problem of time dependency by connecting elements in the ontology that were dependent on time to an element, which described a time interval. For example, a person would be connected to a time interval, which would describe date of birth and death. Solutions for the problem of uncertainty and subjectivity was explored but not implemented. The issue of "why questions" was addressed not in the VICODI ontology itself, but by the system as a whole. To answer a why question, users could use the system to find documents related to the question that might be useful for answering the question.

CIDOC[14] CRM (CIDOC Conceptual Reference Model) is "an attempt […] to achieve semantic interoperability of museum data" [Doerr 2003, p. 76]. Similar to the VICODI ontology, the CIDOC CRM ontology tries to solve problems such as subjectivity and time dependence. As Doerr states, "in history, any conflict resolution of contradictory records is nothing more than another opinion" [Doerr 2003, p. 80] and therefore an

[14] International Committee for Documentation of the International Council of Museums

ontology for the representation of historical "factual knowledge" has to handle those difficulties.

The CIDOC CRM's approach regarding time dependency is to create an event-centric ontology. This means that "temporal entities" and "events" are a central part of the ontology. The actors and objects (physical or conceptual objects) described by the ontology relate to each other through temporal entities. There is, for example, a creation event with properties like start time, end time, and a creator that describes the creation of an object. A creation event could, for instance, describe the creation process of the Last Supper by Leonardo da Vinci. The start time of the event would be 1495, the end time would be 1498, and Leonardo da Vinci would be the creator. [Doerr 2003, Crofts et al. 2011]

Regarding problems such as subjectivity, Doerr acknowledges that the ontology has to be able to capture alternative views. He states that this is part of the design principle: by creating explicit birth events to express when a person was born, it is possible to capture several different "opinions" as alternative.

With their projects, Nagypal et al. and Doerr address problems that are also discussed in the context of the Semantic Web in [Veltman 2004]. Veltman lists five fundamental problems regarding "the history of knowledge organization, knowledge representation and meaning" [Veltman 2004, p. 6]. First, the "pioneers of the Semantic Web" did not consider the possibility of different worldviews or paradigms: a thing is defined by its existence and not by its meaning. This results in the problem that changes in meaning and changing relationships cannot accurately be described. Related to that is the second issue the Semantic Web faces: in the Semantic Web, definition is solely about the existence of a thing; it disregards the fact that there exist other types of definitions (see for

example [Gupta 2008], section 1). Third, Veltman claims that developers of the Semantic Web focus mainly on natural language challenges. They neglect the fact that the same choice of words does not necessarily refer to the same concepts and vice versa. Techniques such as topic modeling are starting to change that situation. However, many applications in the Semantic Web are still based on the assumption that the words of a text alone are enough to conclude its content. Fourth, the possible relationships between two things in the Semantic Web are not expressive enough. Relationships are needed that are able to address issues such as the relationship between concepts and the terms representing them. Last, the Semantic Web does not consider that the meaning of a term can change depending factors such as time, location, or cultural context.

Projects in the digital humanities that want to use the Semantic Web or contribute to it need to solve these issues at least in part (depending on their goals and objectives). As described above, there exist solutions for some problems, such as time dependence. Others might be solvable by developing the right tools (for example, a tool could examine all information regarding a specific term and then decide if this term has more than one meaning based on that information).

Hyman and Renn discuss the fundamental problems of the Semantic Web and other technological approaches for the representation of knowledge in [Hyman and Renn 2012]. They propose the *Epistemic Web* as next step in order to create "a universe of knowledge on the Web that parallels human knowledge" [Hyman and Renn 2012, p. 5]. They suggest that the links between resources on the Web constitute an important kind of knowledge that often stays hidden, but could become autonomous entities of inquiry in an Epistemic Web. Hyman and Renn also critique that in many cases publications only provide parts of the data

that was used to draw a conclusion (such as experimental data or historical sources), and that therefore the verification or reproduction of results can be rather cumbersome. The Epistemic Web is envisioned to solve these and similar problems by providing standards, tools, and openly accessible data contributing to the "globalization of knowledge." [Hyman and Renn 2012]

In this dissertation I describe a system that addresses some of the problems described above in a slightly different way, and might be able to contribute to the Semantic Web to a certain extent. It also provides a method for capturing and analyzing implicit relationships that might be invisible or difficult to detect due to unstructured or distributed data. The system might be a first step towards an Epistemic Web.

2.4 Bioinformatics and Medical Informatics

Bioinformatics and medical informatics are two related but distinct fields. While bioinformatics sits at the intersection of computer science and the life sciences, medical informatics is concerned with the intersection of computer science and medicine (either clinical medicine or biomedical research) [Maojo and Kulikowski 2003]. I believe that these two fields, in regard to certain aspects, can be used as models for the field of digital HPS. This section briefly discusses both fields and examines how specific ideas can be applied to digital HPS.

2.4.1 Bioinformatics

One of the first people to use the term "bioinformatics" was Paulien Hogeweg, a Dutch theoretical biologist in 1970. She (together with Ben Hesper) defined bioinformatics as "the study of informatic processes in biotic systems" [Hogeweg 2011, p. 1]. With advances in

sequencing technology, and thus increasing amount of existing biological sequence data, in the late 1980s researchers started to use the term "bioinformatics" as referring to "the development and use of computational methods for data management and data analysis of sequence data, protein structure determination, homology-based function prediction, and phylogeny" [Hogeweg 2011, p. 3]. To store, manage, and make accessible the rapidly accumulating biological data, the first scientific databases were created. Among them was the Protein Data Bank (PDB) in 1971 and GenBank in 1982 [Attwood et al. 2011]. Together with the advances in computing technology and resources, the need for handling and analyzing biological data contributed to the development of the field of bioinformatics [Attwood et al. 2011].

According to David Lipman (as quoted by Moody) bioinformatics has three major components [Moody 2004, p. 11]:

DISCOVERIES Bioinformaticians make "their own" discoveries that are based on data from biological discoveries.

TOOL DEVELOPMENT Bioinformatics is concerned with the development and sharing of tools.

RESOURCE DEVELOPMENT Bioinformatics is concerned with the development of resources.

Referring back to the projects discussed in section 2.2, similar components can be found in the field of digital HPS. Many of the projects aim to provide new resources, such as digital collections or other digital online resources. Additionally, some projects are concerned with the development of new tools, and a few projects try to make new discoveries using

existing data and research findings. In examining those components more closely, it becomes obvious that digital HPS still has some way to go.

Regarding resource development, digital HPS projects that build online resources are several. However, taking bioinformatics as a model, there are aspects in which digital HPS can improve. For example, an important resource in the field of bioinformatics is the nucleotide sequence database GenBank[15]. By 2010, it contained 108 million individual sequences and over 1000 complete bacteria and archaea genomes [Benson et al. 2010]. GenBank is synched daily with the European Molecular Biology Laboratory Nucleotide Sequence Database (EMBL) and the DNA Databank of Japan (DDBJ) to provide "uniform and comprehensive collection of sequence information" [Benson et al. 2010, p. D46] that can be accessed worldwide. GenBank follows a collaborative approach regarding building the database. New entries are submitted by either individual authors or by sequencing centers, which upload new data in bulk. The data contained in GenBank can be downloaded free of charge via an FTP server or through other web services. [Benson et al. 2010]

When applying the model of GenBank to the field of digital HPS, some similar approaches can be found. For example, the Biodiversity Heritage Library (BHL) aims to digitize biodiversity literature held by different collections and institutions and to make the digitized versions freely available online [Biodiversity Heritage Library 2013]. However, there are two fundamental differences to GenBank. First, BHL can only digitize literature that is out of copyright or for which they have permission from the publisher. This, in many cases, means that newly published works are not included in the digital collection. Second, the data

[15] GenBank can be accessed from the NCBI homepage: http://www.ncbi.nlm.nih.gov/

that BHL provides is mainly unstructured full-text data[16]. Although this is an important resource for historians of science (or for humanists in general), a computer has limited capabilities for analyzing such data, which makes the development of tools to help with the analysis of the provided data much more difficult.

Regarding software development and sharing, bioinformatics has a broad spectrum of tools that can be used by researchers. Wikipedia lists over 50 open-source bioinformatics software tools [Wikipedia Contributors 2014a], which include bioinformatics extensions for programming languages such as Java (called BioJava [Prlić et al. 2012]) or Python (called Biopython [Cock et al. 2009]), and data analysis tools such as Anduril, a software application for analyzing biomedical research data developed by the University of Helsinki [University of Helsinki 2014].

Another bioinformatics application, which has existed for several years, is Cytoscape. Cytoscape is a tool to analyze networks built from biological data, such as protein-protein interactions. It has a plug-in architecture so that it can be customized to the individual needs of researchers. Originally developed by the Institute for Systems Biology in Seattle, Washington, Cytoscape expanded into an open-source community project with a core team of over ten developers from several different institutions (for example, University of California, San Diego or University of Toronto) and a scientific advisory board that directs the development of Cytoscape. Cytoscape is an open-source project using GitHub to host its code that is licensed under the GNU Lesser General Public License, which means that

[16] I am referring to the downloaded text being full text that might be structured into sections and pages but that is no structured data such as a metadata record or as a genome sequence.

anyone can download Cytoscape's source code to use, modify and redistribute. [Killcoyne et al. 2009, Shannon et al. 2003, Cytoscape Consortium 2013]

In comparison, software development projects in digital HPS are not at the same scale yet. As discussed in section 2.2, many digital HPS tools are either developed for a very specific purpose, which does not make it easy to reuse them, or to adapt them to other projects, or the tools are not made available and promoted for reuse. One reason for that is that a big part of the resources used by digital HPS are full-texts or other kind of textual data, so that tools are often tailored to a specific research question and a specific kind of document. Although most of the software applications are open-source, so far there does not seem to be development of any open-source communities with the goal of developing the tools.

The components of resource and tools development and sharing are well established in the field of bioinformatics. Because of that, the third component of making new discoveries based on data from previous biological discoveries is well developed as well. Tools such as Cytoscape support researchers in analyzing large-scale data sets and in generating new hypothesis based on the analysis and exploration of the data that often is provided by resources such as GenBank. Researchers in digital HPS come to new research findings based on data such as digitized sources as well. Projects like the Newton Project use computational tools to find passages in texts that are of interest to researchers and that lead them to new research questions or conclusions. However, digital HPS projects often use very specific resources and tools. This limits the possibilities to find unknown or implicit links between entities, such as documents or people, because the data the tools are using (for example, a specific text corpus) are often carefully selected ahead of time. Also, as projects

often cannot reuse existing resources because, for example, of copyright reasons or because the resources of interest are not yet available in digital form, many projects first need to overcome the problem of collecting digital data before they are able to use tools that might lead to new findings.

Crystallographer Olga Kennard, with the Cambridge Structural Database (CSD), a scientific database that was used as a model to build the Protein Data Bank (PDB) in 1971, fulfilled her dream "to be able to use data collections to discover new knowledge, above and beyond the results yielded by individual experiments" (as cited in [Attwood et al. 2011, p. 6]). If the field of digital HPS continues to develop the three components described above, similar goals might be achievable for digital HPS.

2.4.2 Medical Informatics

Medical informatics is a field with its roots in the 1950s. One of the first applications of computer science toward the field of medicine was the creation of databases for clinical use or as bibliographic references. Other applications include electronic health record systems or medical information systems. An important resource developed in the context of medical informatics is MEDLINE[17], which was established in 1971 by the National Library of Medicine (NLM). MEDLINE is a bibliographic database, containing a total of 19,974,272 citations (as of March 2013, see [U.S. National Library of Medicine 2013a]) from 5,652 journals (according to MEDLINE's own statistics as of November 2013, see [U.S. National Library of Medicine 2013b]). MEDLINE is freely accessible via the Internet. [Maojo and Kulikowski 2003, Collen 1986, Hersh 2002]

[17] MEDLINE (MEDLARS online) is the online version of MEDLARS, which stands for Medical Literature Analysis and Retrieval System. MEDLARS provided digital access to the Index Medicus. [Collen 1986]

The NLM has several other projects contributing to medical informatics. One of them is the Visual Human Project (VHP)[18], which aims to create "complete, anatomically detailed, three-dimensional representations of the normal male and female human bodies" [NLM 2003]. The dataset of digital images provided by the project is intended to act as a standard for the human anatomy to be used in research, education, the development of tools, or other purposes [Ackerman 1998]. To obtain the dataset, a nonfinancial licensing agreement has to be signed and sent to the NLM, which will then provide an account to an FTP server to download the data [NLM 1994]. The VHP dataset has a wide range of applications. For example, it has been used for the simulation of surgeries, as a basis for medical illustrations, and even by car manufacturers for the development of passenger injury models [Ackerman 1998]. The VHP has been followed by similar projects in other countries, for example, the Chinese Visible Human Project [Zhang et al. 2005] or the Visible Korean Human [Park et al. 2006], with the goal to create a 3D anatomic library for education and research [Park et al. 2006]. The ultimate goal of the VHP is the linking of text-based information and images to create a "unified resource of health information" [Ackerman 1998, p. 510] that brings together textual data that for instance describes a disease with medical images that are often necessary for a complete understanding of a disease and human health [Ackerman 1998].

Another project of the NLM is the Unified Medical Language System (UMLS). The goal of this project is to provide a solution for the following two issues it identified: "the variety of names used to express the same concept and the absence of a standard format for distributing terminologies" [Bodenreider 2004, p. D267]. The UMLS aims to accomplish its

[18] See [NLM 2003].

goal by developing several tools (the Metathesaurus, the Semantic Network, the Information Sources Map, and the SPECIALIST Lexicon and Lexical Tools) to provide an extensive biomedical vocabulary that could be used in the development of tools for research, education, and health care [Humphreys et al. 1998, NLM 2011]. The most significant tool is the Metathesaurus [Bodenreider 2004]. It integrates terms from various vocabularies (for example MeSH[19] or SNOMED CT[20]) and links those terms (called "concepts") through a number of relationships such as "is a kind of" or "caused by" [Bodenreider 2004, NLM 2011]. Each concept in the Metathesaurus is identified by a unique id [Humphreys et al. 1998] and links to other relevant concepts, either internally or externally to other databases [Bodenreider 2004]. The UMLS has been used in a number of projects. An example of this is described in Eck et al. when they use UMLS to aid in the translation process of medical texts [Eck et al. 2004]. Another project detailed in an article by Shu et al. ([Shu et al. 2004]) used UMLS for the development of an application to annotate clinical data with "clinically significant events," such as diseases or symptoms.

HPS might benefit from experiences made by medical informatics projects, such as the ones described above, regarding characteristics like collaboration (for instance, UMLS is a highly collaborative project [Humphreys et al. 1998]) or reuse of data (for example, the sharing of the Visual Human Project dataset). Hersh lists four, as he calls it, "core themes" for medical informatics [Hersh 2002], which are also applicable for HPS projects:

STANDARDS As medical informatics is a field that attracted the interest of the industry sector, many software applications have to work with data from different

[19] MeSH (Medical Subject Headings) is a controlled vocabulary administered by the NLM and used in PubMed for the indexing of articles.

[20] SNOMED CT (SNOMED Clinical Terms) is a multilingual, controlled vocabulary of clinical healthcare terms.

vendors or services. To be able to successfully integrate the data, standards play a key role in the field.

TERMINOLOGY Terminology is an example of an area that depends heavily on standards. Computers do not know if two terms refer to the same thing unless they are told so. Hersh gives the example of upper respiratory tract infection, which is similar to a cold. While a human reader makes the connection between these two terms easily, a computer can only do that if there exists a service providing the necessary relationship information for which a standard terminology is often a basic requirement.

USABILITY Many software applications in the field of medical informatics are intended for use in a clinical setting. Therefore, they have to be easily integrated into the daily work routine of physicians or other medical personnel. For example, systems that require a lot of training and effort of the user usually need to result in equally great benefits for the users.

DEMONSTRATED VALUE Similarly to usability, systems usually need to demonstrate that they are of value to its users, otherwise the costs and efforts of introducing the system into the clinical workflow might not be beneficial.

In the context of digital HPS, standards are of increasing importance. The more services there are that provide data for other projects to use, the more important it is for these services to follow common guidelines. For example, a tool for a specific kind of analysis of texts could greatly benefit from services providing texts. However, if each service implements its own standards regarding text encoding and service communication, such a tool would have to be specifically adapted for every service.

Terminology poses a bigger problem to digital HPS than to medical informatics. Digital HPS is not limited to one field, which makes the development of a shared terminology much harder. A vocabulary that would satisfy all kinds of different areas of research could not easily be restricted to a certain number of terms. Another difficulty digital HPS faces is that the same term might have different meanings and definitions depending on factors such as field, time, or place. To develop a common terminology would entail to agreement on a specific meaning of a term, or extending a vocabulary so that it contains all possible meanings of a term. This, however, might not be possible ahead of time, as some meanings might not be known until a specific text corpus is examined. Nevertheless, there are projects such as the Virtual International Authority File (VIAF) that build catalogs of entities such as people, publications, or institutions, and that can be used by digital HPS projects to identify certain "objects."

Regarding usability, I believe that this factor of software applications is as important to tools in medical informatics as it is for those developed in the context of digital HPS. Many HPS researchers have little to no computer science background. Tools requiring users to understand fundamental computer science concepts or that can only be used with extensive training are therefore much more difficult to establish in the community than it might be for bioinformatics tools that are often used by individuals who are already knowledgeable in computer science. Key success for digital HPS tools is therefore that these tools are either very easy to learn and use or that their benefits outweigh the difficulties the tools pose to beginning users.

A similar argument applies to the demonstrated value of software tools in digital HPS. Researchers need to gain some value by using a tool. More complex tools often need

some kind of training before they can be used effectively. A principal investigator of a project might therefore decide against a certain tool because the initial costs for introducing the tool to his group are too high compared to the demonstrated value of the tool.

In 1986, Collen wrote that "[c]omputers, automobiles and telephones are now among the day-to-day tools of physicians" [Collen 1986, p. 778]. While this is certainly true for most historians and philosophers of science, I believe that many have yet to embrace the full potential that computers provide. In 2002, Hersh ends his article [Hersh 2002] by stating that the key challenge for medical informatics is to "develop[] systems that are easy to use and provide demonstrable benefit" [Hersh 2002, p. 1957]. The same is true for any tool in the field of digital HPS.

CHAPTER 3

WHAT TECHNOLOGIES DO WE HAVE?

This chapter will describe the technical background of this dissertation. I will detail concepts closely related to my dissertation topic, such as ontologies or semantic networks. However, the following section will not include implementation-specific technologies, like frameworks, used for developing the software of the Quadriga System or specific file formats. Instead, I will focus on high-level concepts that only in some parts incorporate technical details.

3.1 Authority Files

> *A list of the authoritative forms of the headings used in a library catalog or file of bibliographic records, maintained to ensure that headings are applied consistently as new items are added to the collection.*
>
> — Reitz [2004a, p. 53]

Authority files, which are a component of authority control, are used to organize bibliographic records. An authority file specifies how a person, source, or journal is referenced. An authority file, for example, could define that the physicist Albert Einstein is referred to as "Einstein, Albert, 1879-1955." [Reitz 2004a,b]

Taylor defines in [Taylor 1984] the following terms relevant to the process of authority control:

AUTHORITY WORK Authority work is the process of defining how a name, title, or subject should be referred to in a bibliographic record (usually as heading of that record). It also includes specifying cross-references to other bibliographic records, and how the record relates to other records.

AUTHORITY RECORD An authority record is a record in printed or electronic form that contains the results of the authority work.

AUTHORITY FILE Authority files are "group[s] of authority records" [Taylor 1984, p. 1]. The records grouped by an authority file are also called "entries."

AUTHORITY CONTROL Authority control is the process of using an authority file by referring to its entries with the goal to keep names, titles, and subjects in bibliographic files consistent.

AUTOMATED AUTHORITY CONTROL In contrast to authority control, automated authority control specifically involves the use of software "to manage large portions of the process of authority control" [Taylor 1984, p. 2].

In the time before computers were widely used in libraries, authority records as well as bibliographic records were printed or written on cards. With the turn of the digital age, authority files and bibliographic catalogs were increasingly often built in electronic forms; for example, as databases. With the emergence of the World Wide Web, libraries started to publish those databases online and eventually linked them to other online databases to make them more useful to users.

Often libraries, museums, and other catalog-building institutions create along with a catalog an authority file. For example, the Library of Congress created an authority file[21] to complement its Library of Congress Online Catalog[22]. Similarly, the online catalog of the Deutsche Nationalbibliothek[23] is accompanied by an authority file[24] called "The Integrated Authority File (GND)." Because of the multitude of authority files, there are projects that try

[21] See [Library of Congress 2012]
[22] See [Library of Congress 2013]
[23] See [Deutsche Nationalbibliothek]
[24] See [Deutsche Nationalbibliothek 2013]

to bring together the contents of these authority files in order to provide one interface for all services. Two of these projects are the Virtual International Authority File (VIAF)[25] and Linking and Exploring Authority Files (LEAF)[26].

VIAF is an initiative started by Library of Congress (LC), Deutsche Nationalbibliothek (DNB), and the Online Computer Library Center (OCLC) [OCLC 2014]. The project that was started in 2003 aims to create one international authority file, which includes and merges authority file records from different sources. VIAF provides a web interface that can be accessed at [OCLC 2013b] for human users to search the authority file as well as an interface for software agents[27]. VIAF started out by incorporating only names of people. It was later extended to also include entities such as geographic names, corporate names, or sources. [Loesch 2011]

LEAF was a three-year project, which started in March 2001. Its consortium consisted of about 15 members. LEAF aimed to "enhance search and retrieval facilities by providing high-quality access to international authority information for everybody" [Weber 2004, p. 229]. LEAF's approach was to gather authority records from participating institutions and linking those records in order to create a "Central European Name Authority File." Through this central authority file LEAF would then provide access to the original authority file records. [Weber 2004]

By the end of 2013, VIAF lists about 30 contributors on its website [OCLC 2013a]. Its latest copyright entry is "2010-1013." In contrast, the status of LEAF is not obvious. While the website's copyright entry is "2001-2011," the documents listed on the publications

---

[25] See [OCLC 2013b]
[26] See [LEAF Consortium 2011]
[27] For further information see [OCLC 2010].

webpage are either not delivered or their download links are broken [LEAF Consortium 2004]. The final report of the project is not present as well.

In authority files that are accessible online (such as VIAF) each record often has a unique identifier. The record can usually be retrieved via this identifier through a URL. For example, the authority record for Albert Einstein in VIAF has the heading "Einstein, Albert, 1879-1955" and can be accessed through http://viaf.org/viaf/75121530. The authority record for Albert Einstein in the Library of Congress authority file can be accessed through http://lccn.loc.gov/n79022889.

Closely related to authority files is the concept of "controlled vocabularies" and "thesauri." A controlled vocabulary specifies a limited list of terms that can be used for example when writing a bibliographic record. A bibliographic record typically contains subject terms, which stem from a controlled vocabulary. Using a controlled vocabulary ensures that texts can be found that deal with a certain topic even if they use a different terminology. [Reitz 2004c]

The term thesaurus refers to two different concepts. First, a thesaurus is a collection of synonyms. For example, it states that "residence" or "home" are synonyms for "house." Second, the word "thesaurus" refers to a lexicon of discipline-specific terms. In information retrieval, a thesaurus typically specifies relationships between the contained terms, such as one term being broader or narrower than another term, which allows a user to retrieve either more general or more specific information about a term. [Reitz 2004d]

3.2 Ontologies in Computer Science

> *Ontology in Computer Science, broadly speaking, is a way of representing a common understanding of a domain.*
>
> — Sánchez et al. [2007, p. 3]

In section 2.3 Semantic Web, I briefly described ontologies. In this section, I will describe the concepts and technologies underlying this field in more detail. In the system I describe in this dissertation, the concept of ontologies plays a central role and will be referred to frequently.

Computer scientists adopted the word "ontology" from philosophy, though as explained earlier, it means something different. In the context of philosophy, ontology studies the essence of existence. In the 1980s, researchers in the field of Artificial Intelligence (AI) started to use the term "ontology" in their work. One of them was John McCarthy, an AI researcher, who used the term in his paper on circumscription, which can be described as a "rule of conjecture" [McCarthy 1980, p. 27] that a machine or a person can employ to draw conclusions based on common sense assumptions [McCarthy 1980]. According to Sánchez et al., the concept of ontologies was introduced to computer science because of the need for knowledge representation in artificial intelligence as well as other fields of informatics. Representing the meaning of things rather than merely giving the computer instructions became increasingly important. Ontologies provided a solution to this need. [Sánchez et al. 2007]

A widely accepted definition of ontologies, in the context of computer science, was made by Gruber in 1993. He defines an ontology as an "explicit specification of a conceptualization" [Gruber 1993, p. 1]. Gruber considers a conceptualization to be "an

abstract, simplified view of the world that we wish to represent for some purpose" [Gruber 1995, p. 908].

I will use the broader definition by Sánchez et al. given in the beginning of this section in my dissertation for the following two reasons. My first reason is that Sánchez et al. use the term "domain" rather than talking about a "view of the world." Although Gruber clearly states that ontologies concern only certain areas of interest, using the phrase "view of the world" might give the impression that an ontology tries to describe everything and not just a domain. In fact, it is crucial to understand that an ontology not only represents the "things" in a domain of interest but, moreover, does so according to a common understanding which in some cases might be rather vague. This is the second reason I prefer the definition by Sánchez et al. They include this important detail in their definition. Ontologies are one way to share knowledge between different agents of a system—for example, different computer programs that work with similar kinds of data. To share knowledge, an agreement on the meaning of certain "things" is critical. This agreement is described in an ontology.

In the last paragraph, I avoided being specific about the "things" that an ontology describes. Before I turn to this subject though, I will give a brief example of a scenario in which an ontology can be useful. Consider two computer systems that manage the inventories of two museums. Each museum has exhibits: books, pictures, and sculptures. For a joined project, the two museums now plan to develop a software component that enables the two computer systems of the museums to exchange information about the exhibits. Because the two computer systems were originally developed by different companies for different organizations, the inventories are represented differently. For

example, one museum might have the two categories "painting" and "drawing" in its computer system, while the other one has only one category "picture." If computer system B would ask computer system A for all its pictures, computer system A would not be able to give an answer because it does not know what a picture is, and vice versa. To solve this problem, the two museums could develop an ontology, which describes what they can agree on. The ontology would specify that drawings and paintings are kinds of pictures. If computer system B would now ask computer system A for all pictures, computer system A could look up in the ontology what a picture is, find out that drawings and paintings are pictures, and return a list of all drawings and paintings that it has stored.

The exhibits of the museums are the "things" that the ontology describes in the example. However, ontologies can describe basically anything. Therefore, a general term is needed to name these "things." Often the word "concept" is used for that purpose (see for example [Henderson-Sellers 2012], [Sánchez et al. 2007] or [Gruber 1995]). While for philosophers the nature of concepts is not a trivial issue, computer scientists use the term most of the time without a detailed discussion of its meaning. For example, Sánchez et al. use the Oxford Dictionary definition that describes a concept simply as "an abstract idea" [Oxford Dictionaries 2010, Sánchez et al. 2007]. Other papers, such as [Henderson-Sellers 2012] and [Gruber 1995], don't give a definition at all. [Gruber 2009] avoids using the word "concept" and uses the term "class" instead.

One might argue that what we call the things that an ontology describes does not matter too much in computer science. However, I claim that this vagueness is a symptom of a bigger, underlying problem. Referring back to Sánchez et al.'s definition, an ontology represents "a common understanding of a domain." This means that an ontology describes

the common understanding of concepts. But what is a concept and is there such as thing as a common understanding of the concepts in a domain? If there exist two slightly different understandings of a concept in a certain domain, does this mean there actually exist two concepts? Sánchez et al.'s definition does not take into account that a common understanding of a domain might change over time or might not even exist for certain elements of the domain. The problem is best illustrated using an example.

Consider an ontology of countries and cities. Each country has several cities and one capital. If the ontology is supposed to represent the current political situation, countries and cities can be easily added to the ontology. However, if the ontology is, for example, created as part of a history project and should also provide historical information, it cannot be used with such a simple structure. This is because a country can have several capitals over time, or a city can belong to more than one country. Istanbul, for example, had two other names before: Constantinople and Byzantium; moreover, while it was the capital of the Byzantine Empire, the Ottoman Empire, and the Turkish Republic, it is not the capital anymore today [Encyclopaedia Britannica Online 2012]. The question is how should this be represented in an ontology? Are Istanbul, Constantinople, and Byzantium the same city but with three different names, or are they three different cities that are in the same geographic location? If it were one city with three names, the names would have to have time spans attached that specify when the city had what name. This could become ambiguous when a city has two names that are being used during the same time span. For example, the name "Istanbul" was already used while officially the name Constantinople wasn't changed until 1930 [Encyclopaedia Britannica Online 2012]. In an ontology, it would not be possible to explicitly refer to either Istanbul or Constantinople during the time frame when both names

were in use. On the other hand, if Istanbul, Constantinople, and Byzantium were represented as three different cities occupying the same geographical region, information that stayed the same would have to be entered or changed for each city separately. Moreover, to find out how old Istanbul is, it would be necessary to define rules that specify that in this case the information asked for can be found in Byzantium, not Istanbul.

With the growing interest in the humanities to use ontologies[28], solutions have been developed for problems such as the one described above. In section 2.3 I described two projects concerned with the development of ontologies and the challenges they are facing (CIDOC-CRM and VICODI). Regarding terminology, I will not use the term "concept" in my dissertation to refer to the things that an ontology describes. My main reason for that is that I will use the word "concept" in a different context later on. To avoid confusion, I will therefore use the term "class" in the context of ontologies instead, which is a term that is typically used by computer scientists when discussing the entities of an ontology. I will use "class" to refer to the types of things that an ontology describes. For instance, in the example given above the ontology would have a class "city" and a class "country."

3.2.1 Elements of Ontologies

Ontologies represent "a common understanding of a domain" [Sánchez et al. 2007, p. 3] by defining *classes*, *attributes*, and *relationships* [Gruber 2009]. Classes define the "things" in a domain. Attributes describe the properties a class can have. For example, since countries typically have names, the class country has an attribute name. Relationships specify the

[28] For instance, the First International Workshop on Ontology Based Modelling in the Humanities was held in 2006 at the University of Hamburg (see [Hahn and Vertan 2006]).

relations between classes. For instance, a country has cities. Hence, the class country has the relationship has_cities to the class city.

Classes can have *subclasses*. A subclass has all the attributes and relationships of its so-called *superclass*, as well as optional additional attributes and relationships. For example, the class city could have a subclass village that has an attribute name. Village would be related to country via the latter's relationship has_cities. In addition village could have a relationship next_large_town that specifies the next bigger city close to the village.

In addition to classes, attributes, and relationships, Serrano Orozco describes *axioms* and *instances* [Orozco 2012]. Instances are individual objects that are described by a class. For instance, Istanbul is an instance of the class city and Turkish Republic is an instance of the class country. Axioms are according to Sánchez et al., "sentences that are always true" [Sánchez et al. 2007, p. 11]. They can be used to test if an ontology is consistent [Sánchez et al. 2007]. Moreover, by applying axioms to an ontology "new" knowledge can be derived [Sánchez et al. 2007]. This process is called *inference* or *reasoning*. Extending the country/city example by adding the attributes type and population to the class city, an axiom would be that a city that has less than 100,000 citizens is of type village while a city with 100,000 citizens or more is of type large town.

3.2.2 Ontology-related Terms

There are a few terms that are frequently used in the context of ontology development and usage. This section will briefly describe some of these terms as they might be necessary to understand parts of this dissertation.

Reasoning

Gaševic et al. describes reasoning as "a process of using known facts and/or assumptions in order to derive a conclusion or make an inference" [Gaševic et al. 2009, p. 113]. In computer science, a computer is carrying out the reasoning process (also called inference) by using given information to infer information that is not explicitly stated [de Bruijn et al. 2008]. For example, if the computer "knows" that Istanbul is a city and that each city has a mayor it can infer that Istanbul must have a mayor.[29] The information that the computer is provided with, that it knows, can be described by an ontology. The component that performs the reasoning is called a reasoner or reasoning engine [Hebeler et al. 2009b].

Consistency

The concept of consistency is tightly coupled with reasoning. An ontology is consistent if it satisfies a set of specified rules, which Bloehdorn et al. also call "consistency conditions" [Bloehdorn et al. 2006]. In [Bloehdorn et al. 2006], the authors use a classification from Haase and Stojanovic, which distinguishes three types of consistency: structural consistency, logical consistency, and user-defined consistency. Structural consistency verifies that an ontology is described according to the rules of the chosen ontology language. An ontology is logically consistent if the information described in it is not contradicting. For example, and ontology describing countries and cities might define that a country has exactly one capital. If that ontology contains an instance of a country that has two capitals, the ontology would be logically inconsistent. User-defined consistency is a type of consistency that is concerned with constraints for an ontology that are created by a user and that are not expressed

[29] This is a very simplified example. Most inferences are not that simple. For instance, the case that a mayor suddenly dies and a city does not have a mayor for a certain time span is not taken into account here. However, this simplification demonstrates how complicated a reasoning process can become.

through the ontology language. For example, a constraint could be that if class A is a subclass of class B, and B is a subclass of class C, then the ontology becomes inconsistent if class A is also a direct subclass of class C. [Haase and Stojanovic 2005]

Ontology consistency plays an important role when creating or changing ontologies. An inconsistent ontology might lead to incorrect results or malfunctioning systems. For example, if an ontology specifies two capital cities for a country but the system using that ontology operates under the assumption that there is exactly one capital for a country, the system might fail if it encounters the inconsistent part of the ontology.

Open-World and Closed-World Assumption

Ontologies typically make an open-world assumption. This assumption says "not knowing whether a statement is explicitly true does not imply that the statement is false" [Hebeler et al. 2009a, p. 103]. For example, in an ontology there are two instances of the class city, Istanbul and Berlin, and one instance of the class country, which is Turkish Republic. Turkish Republic is related to Istanbul by the relationship has_cities. Berlin is not related to any class via such a relationship. Under the open-world assumption, the question if Berlin is a city of the Turkish Republic would be answered with "unknown." If we now add another country Germany that is related to Berlin via the relationship has_cities and an axiom that states that each city can only belong to maximal one country, the answer to the question is that Berlin is not a city of the Turkish Republic.

The closed-world assumption, in contrast to the open-world assumption, says that "any statement that is not known to be true can be assumed to be false" [Hebeler et al. 2009a, p. 103]. For the "Istanbul-Berlin-Turkish Republic" example described above, the answer to the question "Berlin is a city of the Turkish Republic" would be answered with

"false" under the closed-world assumption as long as it is not explicitly stated that Berlin is a city of the Turkish Republic.

3.3 RDF

An important concept in the context of the semantic web is the Resource Description Framework (RDF). RDF is a standard created by the World Wide Web Consortium (W3C). It was developed with the goal to "enable[] and promote[] the encoding, exchange, and reuse of structured metadata" [Needleman 2001, p. 58]. It can be described using, for instance, the eXtensible Markup Language (XML) with some added constraints to express the semantics of metadata. [Needleman 2001]

RDF is used to describe relationships between so-called "resources" and to provide information about those resources. Anything with a Uniform Resource Identifier (URI)[30] can be a resource [Gutierrez et al. 2007]. For example, every page in Wikipedia is a resource identified by a URL such as http://en.wikipedia.org/wiki/Digital_humanities. For every Wikipedia page there is also an entry in DBpedia, a project that makes the content of Wikipedia available in a structured, query-able form[31]. Every entry in DBpedia is a resource identified by a URI such as http://dbpedia.org/resource/Digital_humanities.

RDF can be used to create so-called "statements," which are subject, predicate, object triples. A predicate specifies a relationship between a resource, which is the subject,

---

[30] "A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource" [Berners-Lee et al. 2005, §Abstract]. There are two types of URIs: Uniform Resource Locator (URL) and Uniform Resource Name (URN). URLs describe how a resource can physically be retrieved. For example, the URL of a webpage tells the browser what server to contact and what content to request from the server. The primary purpose of URNs, in contrast, is to name a specific resource by being permanently globally unique even if the resource that a URN describes is not physically retrievable. A URI can be both, a URL and a URN, at the same time.

[31] See http://wiki.dbpedia.org/ for further information.

and the object, which can either be another resource or a specific piece of information such as a string of text (called "literal"). For example, an RDF triple could specify that the webpage with the URL http://embryo.asu.edu/pages/thomas-henry-huxley-1825-1895 is about Thomas H. Huxley who is identified by the DBpedia URI http://dbpedia.org/resource/Thomas_Henry_Huxley. Other triples could state that the resource with the URI http://dbpedia.org/resource/Thomas_Henry_Huxley is of type Person and has the name "Thomas Henry Huxley" (see Figure 5). [Rodriguez 2011, Gutierrez et al. 2007]

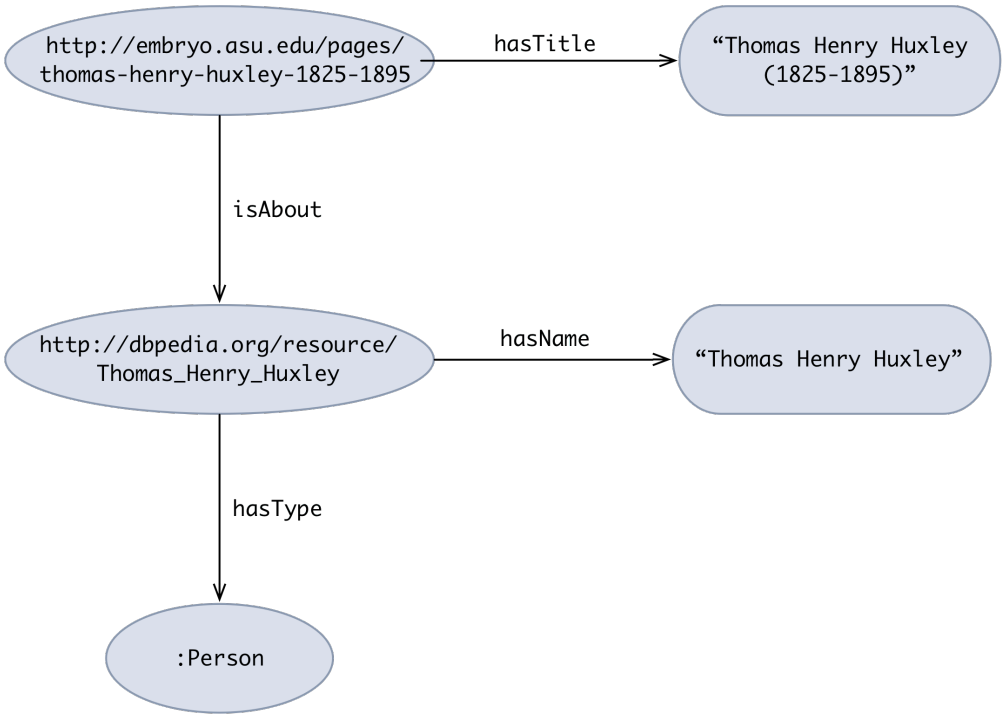

Figure 5: Example RDF Graph: Round Nodes represent Resources and squashed Rectangles represent Literals.

As stated in the beginning of this section, RDF can be expressed using RDF/XML, which is a variant of XML. XML is a markup language, which means that it is used to specify the functions of the different parts of a document [Klein 2001]. For example, using XML

one could specify that the first line of a specific document is the title, or that the second line shows the author of a text (given that this is the case). In XML, the beginning and the end of a part are defined by so-called *tags*. A tag starts and ends with an angle bracket with the name of a tag in-between the brackets, for instance <tagname>. To mark that a tag denotes the end of a part, the tag name is prefixed with a slash: </tagname>. Listing 1 shows a simple example XML document that describes an address.

Listing 1: Simple XML Example

```
1  <address>
2    <name>Julia Damerow</name>
3    <street>Maple Street</street>
4    <city>Phoenix</city>
5    <state>Arizona</state>
6  </address>
```

RDF/XML defines a specific structure for RDF documents. For example, an RDF document starts with a tag <RDF> followed by all statements of the document. Listing 2 shows an example RDF document that describes the RDF document shown in Figure 5. Lines two and three contain so-called *namespace declarations*, which can be understood as vocabulary definitions. For example, the RDF namespace contains specific tag names relevant for RDF. Lines five to eight contain the description of the Embryo Project webpage about Thomas Huxley (line six defines its topic, line seven defines its title). Lines ten to thirteen describe the resource representing Huxley in Dbpedia.

Listing 2: Simple RDF Example

```
1  <rdf:RDF
2        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3        xmlns:ex="http://example.org/">
4
5       <rdf:Description rdf:about="http://embryo.asu.edu/pages/thomas-henry- huxley-
             1825-1895">
6              <ex:Topic rdf:resource="http://dbpedia.org/resource/
                    Thomas_Henry_Huxley" />
7              <ex:Title>Thomas Henry Huxley (1825-1895)</ex:Title>
```

```
8       </rdf:Description>
9
10      <rdf:Description rdf:about="http://dbpedia.org/resource/ Thomas_Henry_Huxley" >
11              <rdf:Type rdf:resource="http://example.org/Person" />
12              <ex:Name>Thomas Henry Huxley</ex:Name>
13      </rdf:Description>
14
15  </rdf:RDF>
```

Namespaces point to RDF vocabularies, which are called *schemas* in RDF. Schemas are defined using RDF Schema (RDFS). They define the elements (the tags) that can be used in an RDF document and how these elements relate to each other. [Powers 2003a]

RDFS can be used to create ontologies. However, its expressiveness is limited. For example, it is not possible to define that if a person pays a monthly rent for a house, there cannot be a relationship defining the monthly mortgage for the same house. For that reason, ontology languages were developed: one of which being the Web Ontology Language (OWL)[32]. [Powers 2003b]

OWL is defined using RDFS and can therefore be used in any RDF document. Using OWL it is possible to develop much more precise ontologies than with RDFS only, as OWL provides more detailed constraints by, for example, restricting the type of properties, or how often they can exist. [Powers 2003b]

Another technology closely related to RDF is SPARQL. SPARQL can be used to "query and manipulate RDF graph content on the Web or in an RDF store" [W3C 2013a, §1]. It is a widely used standard by the W3C (see [W3C 2013a]). Besides defining a query language that allows developers to query RDF data, it also specifies the protocol for such

[32] The "correct" order of the characters in the acronym for Web Ontology Language would be "WOL." However, the creators of Web Ontology Language (OWL) decided to use "OWL" for several reasons. In an email thread discussing possible acronyms, Tim Finin wrote, "(1) it [OWL] has just one obvious pronunciation which is easy on the ear; (2) it opens up great opportunities for logos; (3) owls are associated with wisdom; (4) it has an interesting back story." [Finin 2001]

requests [W3C 2013a, Hebeler et al. 2009c]. For example, Listing 3 shows a SPARQL query for the RDF example above that asks for the titles of all resources.

Listing 3: Simple SPARQL Query

```
SELECT ?title
WHERE { ?resource ex:Title ?title }
```

Listing 4 in contrast shows how a SPARQL query should be sent to a server according to the SPARQL protocol. A service that can answer such SPARQL requests is called a SPARQL *endpoint*.

Listing 4: SPARQL Protocol Example

```
GET /sparql/?query=PREFIX%20ex%3A%20%3Chttp%3A%2F%2Fexample.org%2F%3E
      %0ASELECT%20%3Ftitle%20%0AWHERE%20%7B%20%3Fresource%20ex%3ATitle
      %20%3Ftitle%20%7D HTTP/1.1
Host: www.myexample.org
User-agent: my-sparql-client/0.1
```

RDF data is often stored in so-called *RDF stores* or *triple stores* [Hebeler et al. 2009b]. These stores are specifically developed for RDF subject, predicate, object triples. Typically, they also include query engines that handle requests in SPARQL format for stored RDF data made by developers (or applications) [Allemang and Hendler 2009]. A special characteristic of triple stores is that in contrast to relational databases they are able to easily merge datasets [Allemang and Hendler 2009]. A dataset A that is the result of merging dataset B and C contains all the triples in dataset B, as well as all the triples in dataset C, and two resources are equivalent if they are identified by the same URI [Allemang and Hendler 2009]. In a relational database that consists of tables (similar to an Excel Spreadsheet), the merging of

two tables would be much more difficult as it is not generally defined what columns are equivalent. Figure 6 illustrates that difference with an example[33].

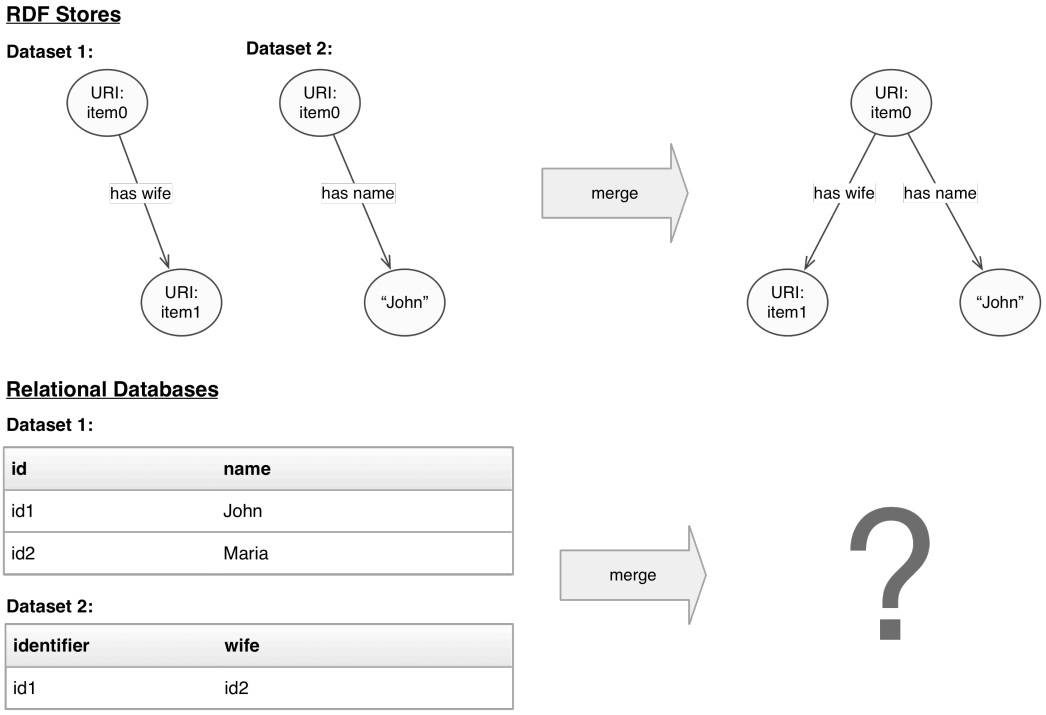

Figure 6: Merging of Data in RDF Stores versus Merging of Data in Relational Databases

3.4 Linked (Open) Data

In 2001, Berners-Lee et al. envisioned that "[b]y augmenting Web pages with data targeted at computers and by adding documents solely for computers, we will transform the Web into the Semantic Web" [Berners-Lee et al. 2001, p. 36]. Five years later, he noted in [Berners-Lee 2006] that a lot of data sets are still not available or linked and he proposed a set of best practices for publishing data in the Semantic Web to create a "single global data space"

[33] Datasets in relational databases can of course be merged as well, however, if column titles are not labeled in structured way (for instance, all columns containing identifiers are labeled "id") the merge process requires an extra mapping process in which equivalent columns are identified.

[Bizer et al. 2009, p. 2]. He called data that was made available following those best practices "Linked Data." [Bizer et al. 2009, Berners-Lee 2006]

Berners-Lee suggested that Linked Data should be published using the following four principles [Berners-Lee 2006]:

1. The "things" documents describe should be referred to using URIs.
2. Those URIs should be HTTP[34] URIs, for example http://dbpedia.org/resource/Thomas_Henry_Huxley. This way, the things referred to by URIs can be looked up on the Internet using for example a Linked Data browser (similar to a web browser).
3. If a URI is looked up, "useful information" should be provided using standards such as RDF.
4. The provided information should include other URIs that link to related information.

Many services or institutions develop software applications that make their data accessible through *Web APIs*[35]. Those Web APIs enable other programs to retrieve data. Berners-Lee developed the idea of Linked Data because he recognized that although data was put on the Semantic Web (for example, by such Web APIs) it was often not linked [Berners-Lee 2006]. In addition, many Web APIs provided their data in their own format, so that only programs that knew the format could use the data [Heath and Bizer 2011]. A large amount of data was therefore not discoverable by Semantic Web agents. As Heath and Bizer phrase it, "while Web APIs make data accessible *on the Web*, they do not place it truly *in the*

[34] Hypertext Transfer Protocol (HTTP)
[35] Application Programming Interface

Web, making it linkable and therefore discoverable" [Heath and Bizer 2011, p. 3, original emphasis]. Berners-Lee's best practices (or has he called it, "rules") suggested a way to standardize and connect different datasets and to thereby create a "Web of Data," in which it is possible to discover information by following links provided in the descriptions of resources. [Bizer et al. 2009, Berners-Lee 2006, Heath and Bizer 2011]

In 2007, the Linking Open Data (LOD) project started. This project has the goal to make "data freely available to everyone" [W3C 2013c, § Project Description] and to publish datasets that are already freely available (such as Wikipedia or GeoNames) according to the Linked Data guidelines. The diagram in Figure 7 shows how different datasets in the LOD project are connected to each other. If two nodes in the diagram are connected, then there are at least 50 links between the two datasets. The size of a node is an indicator for the size of the corresponding dataset. The color of a node corresponds to the category of a dataset (for example geography or life sciences). [W3C 2013c, Bizer et al. 2009, Cyganiak and Jentzsch 2011]

The Web of Data has many similarities with the "traditional web." For example, one of the most significant features of the Web of Data is its connectivity. Users can get from one resource to another by following links to related information. Furthermore, anyone can add data to the traditional web by creating new webpages and websites that can be accessed using a web browser. Similarly, anyone can publish new RDF documents following the Linked Data guidelines that then can be browsed using a Linked Data browser. According to Bizer et al., in August 2011 the Linking Open Data project consisted of 295 datasets that amount to over 31 billion triples. [Yu 2011]

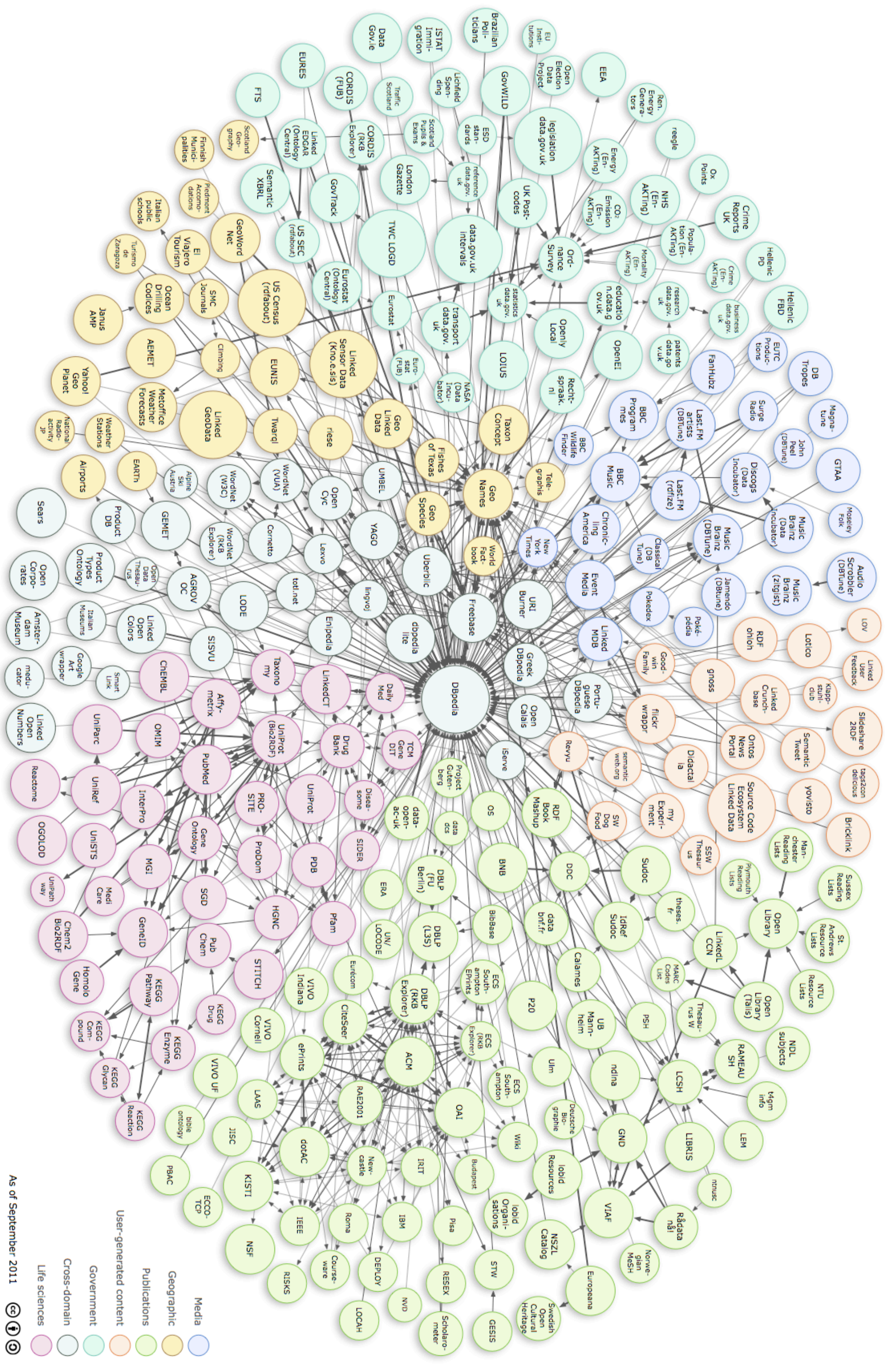

Figure 7: Linking Open Data Cloud Diagram as of September 2011, by Richard Cyganiak and Anja Jentzsch. http://lod-cloud.net/

3.5 Quadruples and Named Graphs

RDF and its triple logic is the basis for the Semantic Web and is widely used. However, it comes with certain drawbacks that create problems in specific situations. An issue often encountered is that a triple does not carry enough information; in many cases an application needs additional information *about* a triple to function properly [Carroll et al. 2005, Macgregor and Ko 2003]. For instance, it is often necessary to know where a piece of information came from. An application that collects triple data from different places needs to track the provenance of data to decide, for example, if it is trustworthy. Another reason for attaching additional information to triples could be access restrictions (not every user can access all triples), or to express an assertion about a triple (such as that the statement represented by a triple is wrong). [Carroll et al. 2005]

A widespread solution for this problem is the use of so-called *quadruples* or *quads* [Macgregor and Ko 2003]. In contrast to triples, which consist of subject, predicate, and object, quads have a fourth element that is often called a *context* (context, subject, predicate, object). This context can contain simply an identifier, which can be used to make assertions about the enclosed triple, or it can point to another resource that contains information about the context [Carroll et al. 2005]. For example, in [Macgregor and Ko 2003], the authors illustrate how quads could be used to query for all the ships that were located in a particular harbor at a specific time and that carried aluminum pipes. When expressed in triple format this is a rather difficult task. Attaching place, time, as well as cargo information to a ship would make time a property of a ship and not of its "situation." It wouldn't be possible to

store the information that a ship anchored in several harbors at different times[36]. If the data were instead to be stored in a quadruple format, in which the forth element pointed to a context, then time information for ships could be stored in the context while information about the cargo of the ship and where it anchored could be attached to the ship (see Figure 8).

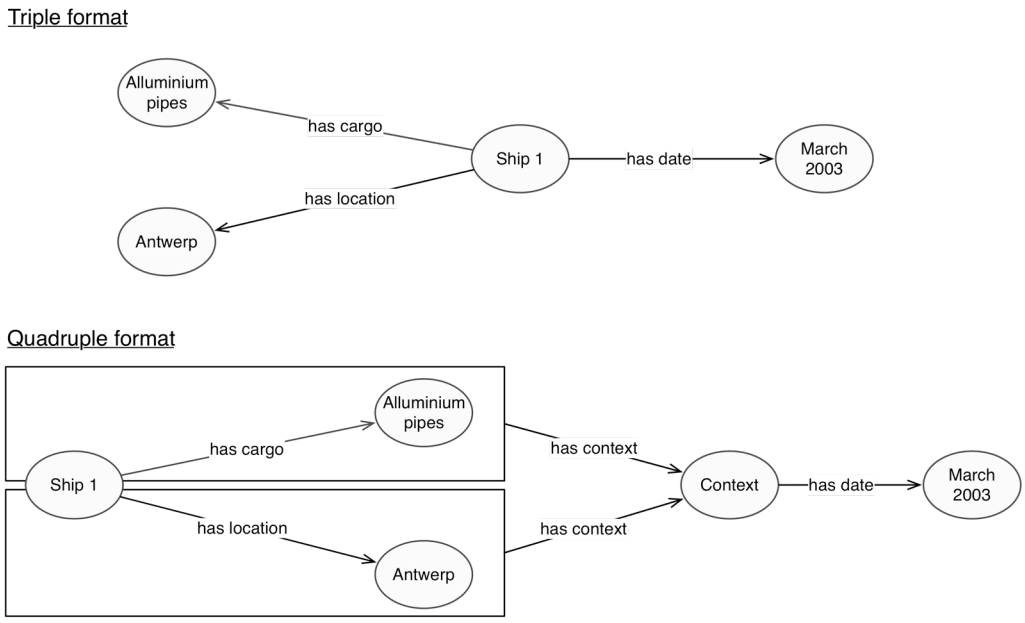

Figure 8: Quads versus Triples

[36] It is possible to express the example in RDF, however, a technique called *reification* is required, which essentially turns a triple into four triples and is rather problematic [Carroll and Stickler 2004].

Another approach to solve the problem described above are *Named Graphs*. The concept of Named Graphs (as proposed in [Carroll et al. 2005]) assigns each RDF graph a name. This name is a URI that can be used to make assertions about a graph. In contrast to other approaches that include contextual information in RDF graphs such as [Guha et al. 2004], Carroll et al. state that Named Graphs are advantageous because they require only minimal changes to the underlying RDF model and can therefore be used with existing tools.

3.6 Semantic Networks and Knowledge Graphs

According to Sowa, a "semantic network or net is a graph structure for representing knowledge in patterns of interconnected nodes and arcs" [Sowa 2013, §1]. Nodes represent, for example, ideas, events, or objects; and edges (or arcs) indicate relationships between them [Lehmann 1992]. The history of semantic networks goes back to Porphyry, a Greek philosopher, who in the third century created the first known semantic network as a response to Aristotle. His network depicting Aristotle's Categories became known as the *Tree of Porphyry*. Semantic networks have been developed by many different scholars in fields such as philosophy, linguistics, or artificial intelligence. [Sowa 2013]

In [Sowa 2013], the author distinguishes six categories of semantic networks:

DEFINITIONAL NETWORKS This type of networks focus on is-a hierarchies that specify "superclass" and "subclass" relationships. For instance, such a network could define that Pluto is a dog and a dog is an animal. Dog would be a superclass of Pluto, and a subclass of animal. Definitional networks also include inheritance, which means that the properties of one element are inherited by its subclasses. For example, because dogs have tails, Pluto has a tail as well. Figure 9 shows an

example of a simple definitional network. Note that the specific syntax of how semantic networks are drawn varies.

ASSERTIONAL NETWORKS These networks represent assertions. For example, an assertional network could express that dogs that have a tail bark or that Daisy says that Pluto has a tail.

IMPLICATIONAL NETWORKS This kind of network focuses on representing implication statements such as if Pluto is wet he took a bath or got in the rain.

EXECUTABLE NETWORKS An executable network has elements that can lead to changes in the network. An example is a *dataflow graph* that can for instance represent a mathematical formula. Dataflow graphs describe how the content of specific nodes is calculated based on some input values.

LEARNING NETWORKS Networks that belong to this type "learn" based on new information. They adjust their weights or simply expand according to new data. *Neural nets* belong to this class of networks.

HYBRID NETWORKS Networks that fall into several of the above categories are hybrid networks. They define the structure of data, as well as how assertions can be made using the data. Although Sowa labels this type hybrid network, he broadens the term by using "hybrid system" instead when explaining this type. A hybrid system can consist of several components that define their own syntax; for instance there is one type of syntax for specifying the structure of data and a different syntax for creating rules about the data. Sowa notes that networks that combine definitional and assertional networks but use the same syntax for both types are usually not considered hybrid networks.

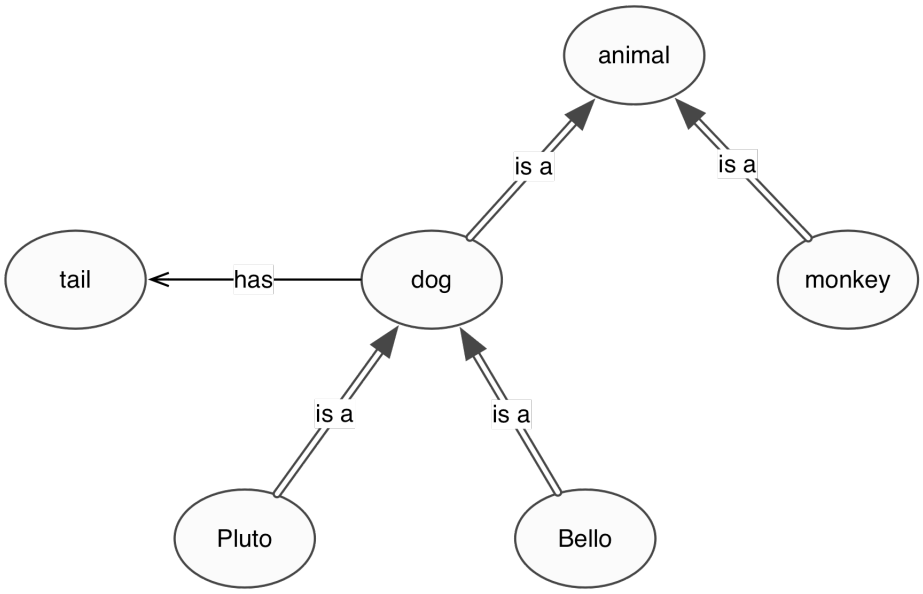

Figure 9: Example of a simple Definitional Semantic Network

In 1976, Sowa proposed a type of semantic network called *conceptual graphs*. The basic units of these graphs are called *concepts* that are connected by *conceptual relations* to form networks. Concepts can be anything: "an entity, action, or property in the real world, an abstraction, fantasy, or mathematical function" [Sowa 1976, p. 337]. Each concept is identified by a so-called *sort label*, which are shown as labels of the nodes of the conceptual graphs. Figure 10 shows a simple conceptual graph representing the statement "a boy walks." [Sowa 1976]

Figure 10: Example of a Simple Conceptual Graph as shown in [Sowa 1976, p. 338]

Conceptual graphs can be ill defined based on so called "formation rules" that pose constraints on graphs. Such rules define, for instance, hierarchies of concepts such as " a lion is an animal," or they define what actions can be attached to what entities, such as "only animals can sleep" but not ideas. Furthermore, Sowa introduces so-called *conceptual schemas*, which can be used to map queries to a database. For instance, a conceptual schema could represent a query such as "what are all the parts each supplier stocks." Sowa envisioned conceptual graphs as a way of translating natural language queries into executable database queries. A computer could parse a natural language sentence, transform it into a conceptual graph, to then use that graph to run necessary database processes. [Sowa 1976]

Semantic networks are used in a variety of fields. They are often employed in artificial intelligence and information extraction projects (for instance [Román et al. 2012]). Sowa developed conceptual graphs as an interface for databases. However, they are also of interest to the humanities. In the context of history of science, one project by Malcolm Hyman reported in [Hyman 2007] employed semantic networks to study semantic change. I will describe this project in more detail as it closely relates to the topic of this dissertation.

In [Hyman 2007], the author describes an approach to create semantic networks using Latent Semantic Analysis (LSA), a computational method to calculate how closely related two terms are. LSA is based on the assumption that if terms are closely related then they frequently appear together in the same chunk of text (a chunk can be a whole document, paragraph, or sentence, depending on the specific problem LSA is applied to). Terms that are closely related, which means that their similarity is above a specific threshold, are connected in the semantic graphs. [Hyman 2007]

Hyman states that a concept can have several expressions by which it is represented. For example, a person can be called by their first, last, or pet name, but it always refers to the same person. In contrast, a term can have several meanings, referring to different concepts. Hyman gives the example of the term "force," which can refer to the concept of force in the field of physics or to a body of military or police personnel such as in "police force." Citing [Kintsch 1998], Hyman asserts that what a term means can best be defined by taking into account the other terms or concepts (Kintsch uses the term "nodes") that are used along with the term. Semantic networks therefore present a useful tool to understand a scientific text's terminology and to study conceptual change. Hyman created semantic networks for parts of the *Problemata Mechanica* to study the vocabulary used around the term "force." This project was a first step in understanding how the "mechanical world-view" changed between Aristotle and Newton. The next step would broaden the text corpus to commentaries on the original text and use semantic networks in order to explore the conceptual change leading to Newton's *Principia* in 1687. [Hyman 2007]

In the context of search engines and semantic search, semantic networks are often referred to as *knowledge graphs*. Knowledge graphs are typically definitional networks that store information to better answer a user's search query. For instance, Google's search engine includes a knowledge graph that shows pictures painted by Da Vinci when searching for "Da Vinci."[37] Kasneci et al. describe a system called "NAGA" that, too, is built on a knowledge graph that they extracted from various sources such as Wikipedia or IMDB [Kasneci et al. 2008]. Their knowledge graph has about fourteen million statements that use

[37] See http://www.google.com/insidesearch/features/search/knowledge.html for Google's knowledge graph demonstration.

90 relationships such as "is a" or "politician of" [Kasneci et al. 2008]. Users can browse or query the knowledge graph to ask questions such as "who are all the physicists born in the same year as Max Planck?" [Kasneci et al. 2008]. A detail that Kasneci et al. do not mention is whether or not a user can find out where the information of the answer has come from. It does not seem to be the case that NAGA attaches provenance information to their "facts." In [Notess 2013], the author critiques Google's knowledge graph for the same issues. A user cannot find out where Google retrieves the information for a fact like that Da Vinci painted the Mona Lisa. This poses a big problem, as provenance information is necessary for decisions about what information is correct in cases of conflicting data, and to establish trust in search results.

CHAPTER 4

A QUADRUPLE-BASED RESEARCH SYSTEM

> *As many fields become overwhelmed with information, it will become physically impossible for any individual to process all the information available on a particular topic.*
>
> —Fan et al. [2006, p. 77]

This quote by Fan et al. describes a problem that an increasing number of research fields are facing. The relevant information available to scholars becomes too much to be analyzed by a human alone, and computers are increasingly employed to aid in the analysis process. In many natural sciences, a big part of the relevant data is at least partially structured or has a numerical format, which makes the computational analysis of such data possible or at least easier. In the history of science, researchers usually use unstructured data such as texts or images. Such data first has to be transformed into a machine-understandable format before it can computationally be analyzed.

The research system I am proposing (the "Quadriga System") is based on the idea of representing texts as graphs, which can be transformed into knowledge graphs. By representing unstructured texts as graphs, the information contained in a text is given a mathematical structure that can be used for computational analysis. However, the underlying assumption of the system is that it would not be possible to capture all knowledge (or even just parts of it) in one knowledge graph[38]. This is especially obvious in the history of science, which is a discipline that studies conceptual change and is therefore operating on the assumption that every two "things" in the world (in the following referred to as

[38] I am explicitly avoiding defining what knowledge is and what the phrase "all knowledge" could refer to. An important part of the system is that what is important information that should be incorporated into a knowledge graph depends on the person examining the information and therefore cannot be generally defined.

"concepts"[39]) have a relationship that is continuously changing. To accommodate such variability, the system I am proposing is built on the idea that any statement can only be analyzed and interpreted in the context that the statement was made. Applied to knowledge graphs, this means that any relationship between two nodes can only be fully understood if it is known based on what information a relationship was created. In the Quadriga System, the context of a statement consists of the text the statement appears in, the text's metadata, and information about the researcher reading and interpreting the text.

The Quadriga System has several components that support the following workflow. A researcher annotates each text of interest with a graph that represents their interpretation of that text. Such a graph consists of relationships between concepts that the researcher created according to the relevant statements of a text. Relevant information is, in this context, the information that the researcher classifies as being relevant. Next, additional information such as metadata of the text is attached to the graph. The researcher then uploads his graphs to a common repository. This repository holds graphs from several researchers working on possibly different projects. Once his graphs are uploaded, the Quadriga System enables the researcher to analyze them, incorporating or excluding specific graphs created by other researchers and projects. In the following, the term "annotator" refers to a researcher reading and annotating a text. "Graphs," "networks," or "annotations" refer to the resulting graphs that researchers create.

In the following section (What is a Quadruple?), I will describe the basic data structure on which the Quadriga System is built. In sections 4.2 and 4.3, I will then detail

---

[39] I am using this term in its broadest sense, simply as "an abstract idea" as defined by [Oxford Dictionaries 2010].

how concepts are used in the system. In section 4.4, I will describe how networks are visualized in the Quadriga System. Section 4.5 will explain how ontologies can be used in the system. Section 4.6 and section 4.7 will describe the system architecture and some implementation details.

4.1 What is a Quadruple?

In section 3.3 (RDF), I described the concept of subject, predicate, object triples. Section 3.5 (Quadruples and Named Graphs) details how this idea has been extended to quads and named graphs. The research system I am proposing is based on so-called "Quadruples," (see Figure 11) also referred to as "contextualized triples," which are quads with a structured fourth element[40].

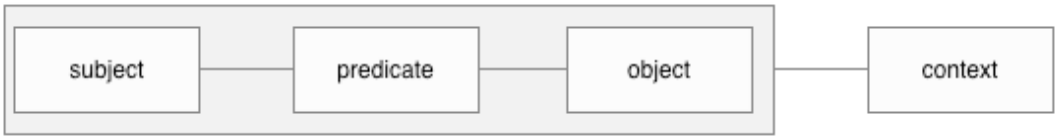

Figure 11: Structure of a Quadruple

The basic idea is similar to an idea proposed in [Macgregor and Ko 2003]. Macgregor and Ko describe quads in which a context can be part of a set of assertions that define the "environment" of that context. Statements made in such a context are considered to be true in the environment of the context. However, Macgregor and Ko do not define a structure for environments. In the case of Quadruples in the Quadriga System, the context is well-defined. It consists of three parts:

[40] To avoid confusion, I will call the concept of quadruples as they are described in, for example, [Dumbill 2003] "quads." In contrast, I will call the concept I am proposing "Quadruples."

METADATA OF RESOURCE The context points to the metadata of the resource that was annotated, such as publication year, author, or publisher.

ANNOTATION CONTEXT This part of the context specifies who annotated a resource, when was the resource annotated, and what institution did the annotator belong to.

CREATION CONTEXT The creation context describes who uploaded the annotation to a shared repository.

The Quadruple concept is similar to the concept of "nanopublications" as proposed in [Groth et al. 2010]. Nanopublications consist of subject, predicate, object statements that have information attached (so-called "annotations") such as the creator and creation date of a statement. All nanopublications that contain a certain statement are called "S-Evidence" for that statement. A reader of nanopublications (either human or machine) can use the S-Evidence and annotations to decide, for instance, between two conflicting statements, or to select only statements of certain creators for their work. [Groth et al. 2010]

The structure of nanopublications, however, is still different than a Quadruple's structure. Also, the description of Quadruples as a variant of quads is a simplified perspective on Quadruples. Quadruples represent several interpretation actions of a researcher and are accordingly more complex than quads. First, a researcher interprets that certain terms in a text refer to certain concepts in the world. For example, the reader of the sentence "Einstein was a physicist" interprets that "Einstein" refers to Albert Einstein, the creator of the theory of relativity, and that "physicist" refers to the profession of researching physics. These interpretations are sometimes trivial, but often they are not. Second, a researcher interprets the meaning of a statement and transforms this into one or several

relationships between concepts. For the example sentence above, a researcher could create the relation <Albert Einstein - is - physicist> in which all three relationship elements refer to the interpretations that assign meanings to the terms in the sentence.

In the Quadriga System, these two layers of interpretation (what do terms in a text mean and how to they relate to each other) are represented by "Appellation Events" and "Relation Events." These are types of nodes that are connected and form graphs that represent interpretations of texts. Such graphs can be computationally analyzed.

4.1.1 Appellation Events

In [Macgregor and Ko 2003], contexts themselves can be subjects of quads to describe the environment of a context[41]. A context therefore consists of several parts: an identifier and an environment that is composed of several assertions that describe the properties of the environment. For example:

```
Identifier: ctx1
Environment:
<ctx1 - hasYear - "2013" - no context>
<ctx1 - hasPlace - "Arizona State University" - no context>
```

Because a context consists of several parts, I will call it a "complex object." In contrast, predicate, and object of the relation <ctx1 - hasYear - "2013" -no context> are "simple objects" in Macgregor and Ko's system as they do not consist of several parts.

In a Quadruple, subject, predicate, and object are of type Appellation Event, which is a complex object with several properties that describe by whom, and when a term was annotated and how the term was interpreted.

---

[41] Usually, quads in which the subject is a context have empty contexts.

DEFINITION 1 An Appellation Event describes the event that a researcher assigns meaning to a term in a text.

Appellation Events have several parts that have different purposes. They describe the term that has been annotated in the text, as well as the meaning of the term, and the context of the annotation, which has the same structure as the context of a Quadruple described above. Table 6 shows a list of all the properties of an Appellation Event.

Table 6: Properties of Appellation Events

PROPERTY	DESCRIPTION
Context	The context describes what source was annotated, who assigned a meaning to a term, and who submitted the annotations to a repository.
Annotated term	This property describes what term was annotated. For example, in the sentence "Einstein is a physicist" the term "Einstein" would be annotated. The annotated term can have several parts (e.g. "Albert Einstein") and the different parts don't have to be connected (e.g. if in the sentence "the man on the other side of the street with a beard" we want to annotated "the man […] with a beard").
Normalization	The normalization contains the grammatical normalization of a term such as singular of plural nouns, or infinitive forms of verbs. For example, for "Einstein likes horses" "likes" would be normalized to "like" and "horses" would be normalized to "horse."
Referenced terms	In cases in which words (such as pronouns) refer to another word in the text (anaphora) this property contains the referenced term. For instance, in "Einstein was a physicist and many people know him" "him" refers back to "Einstein." However, this property should only be used if it is certain from the grammatical structure of the sentence to what other term a word refers. If this cannot be inferred from the grammatical structure and is therefore up to the interpretation of the annotator, then this property should not be used.

PROPERTY	DESCRIPTION
Interpretation	This property describes the meaning of a term. For example, in the sentence "Einstein is a physicist" the Appellation Event for "Einstein" would describe that here the physicist and creator of the theory of relativity is meant.
Certainty	In some cases the annotator might not be absolutely certain about the meaning of a term. In these cases, this property can be used to indicate such a situation.

4.1.2 Relation Events

Like Appellation Events, Relation Events are complex objects. They describe by whom and when there was stated that there exists a relationship between two elements in a text. In the simplest case, they consist of three Appellation Events and a context. However, the subject or object of Relation Events can also contain other Relation Events.

DEFINITION 2 A Relation Event describes the event that a researcher infers from a text that there exists a relationship between two elements. These elements can either be concepts or relationships between concepts.

Typically, the term "Quadruples" refers to Relation Events. They have a subject, predicate, object, and a context. Subject and object can either be Appellation Events or they can refer to other Relation Events. In contrast, the predicate of a Relation Event is always of the type Appellation Event.

Table 7 shows the properties of a Relation Event.

Table 7: Properties of Relation Events

PROPERTY	DESCRIPTION
Context	The context describes what source was annotated, who created the annotations, and who submitted the annotations to a repository.

PROPERTY	DESCRIPTION
Subject	The subject of the relation being created. This can either be an Appellation Event or a Relation Event.
Predicate	The predicate of the relation being created. The predicate has to be of the type Appellation Event.
Object	The object of the relation being created. This can either be an Appellation Event or a Relation Event.
Relation creator	The creator of a relation is in most cases the researcher annotating a text. However, some texts describe that a specific person made a statement about a topic. In such cases, it might be useful to store that information in the corresponding Relation Event. For example, in the sentence "Einstein said that Bohr is a physicist" Albert Einstein would be the relation creator in the Relation Event that describes the relation <Bohr - is - physicist>.

A Relation Event that refers to another Relation Event in its subject or object is called a nested Relation Event (or a nested relationship). Nested Relation Events can be used to express complex statements that can't be formulated as triples or to make assertions about relationships. For example, the statement "Arizona has a population of 6.5 million" consists of two assertions: Arizona is a population, and that population is 6.5 million. If the sentence would be represented as two separate relationships, as shown below in Listing 5, the connection between the two triples would be lost.

Listing 5: Relationships without Nesting

```
<population - is - 6.5 million>
<Arizona - has - population>
```

Moreover, the first relationship would simply state that there is a population of 6.5 million, which is a worthless piece of information unless we are interested in what numbers

of populations exist. For situations like that, nested Relation Events are needed. In a nested relationship, the second relationship can refer to the first relationship. This way the sentence is represented correctly (Listing 6):

Listing 6: Relationships with Nesting

```
R1: <population - is - 6.5 million>
R2: <Arizona - has - R1>
```

All Relation Events that belong to one nested Relation Event (in the example above the two Relation Events representing the shown relationships) are called a *statement* in the Quadriga System. A statement usually represents a claim in a text that an annotating researcher considers relevant[42].

4.1.3 What's with all the contexts?

Because Relation Events and Appellation Events are understood as separate interpretation actions, the two types of objects have separate contexts. This design of the structure of Quadruples has specific implications for the Quadriga System and how researchers can use it.

First, separate contexts for Appellation Events and Relation Events mean that Relation Events can be created using Appellation Events and Relation Events created by another researcher. This way, researchers can make statements about the interpretations of other researchers. Figure 12 illustrates such a scenario. Researcher A (solid filled nodes) created a Relation Event using three Appellation Events. For example, the Relation Event could express that <Einstein - was born in - 1880 - context A>. Researcher B (striped nodes)

[42] The word "claim" is used in a very broad sense here. It could be an actual claim the author of a text made in the text, or an indirect statement such as the acknowledgment of a colleague (which could be represented by a "knows" or "acknowledges" relationship between the two people).

then created a Relation Event using Researcher A's Relation Event and two Appellation Events that he created himself. For example, Researcher B could state that < <Einstein - was born in - 1880 - context A> - is - wrong - context B>. Who the creators of the Relation Events were and when they were created would be specified in context A and context B.

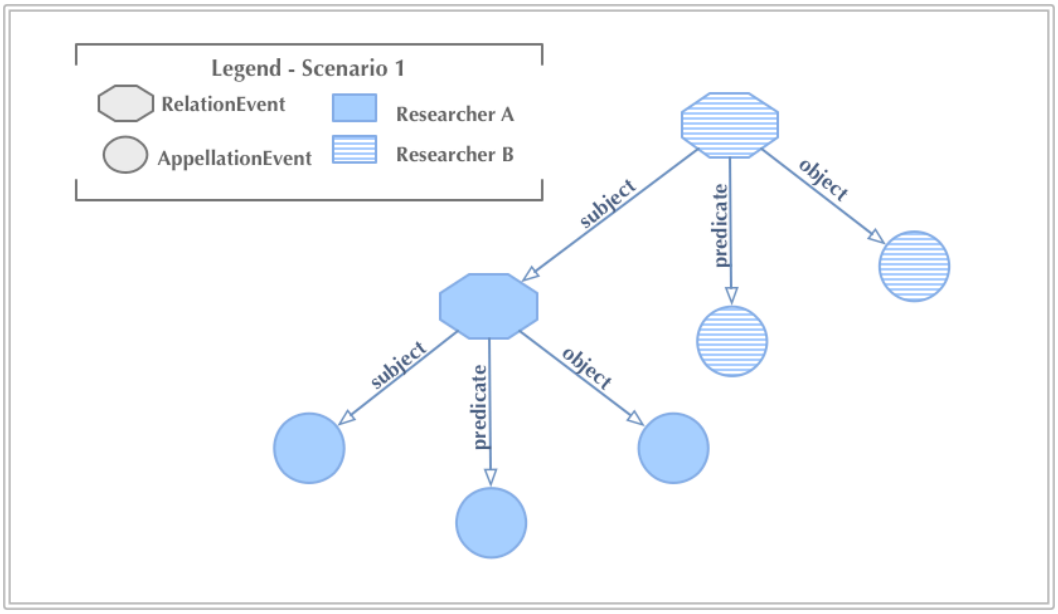

Figure 12: Using a Relation Event from a Different Annotator

Second, because every node in the graphs that are created by the Relation Events and Appellation Events contains information about when, where, by whom, and for what resource it was created, this information can be used to search and filter these graphs. For example, if a researcher analyzing the graphs is only interested in annotations created by a specific person, the nodes can easily be filtered. The same is true if only texts from a specific time period or by a specific author are of interest. Figure 13, scenario two shows two graphs that were created for two different texts but that both use the same two concepts (nodes that are half blue and half striped). Using the graphs' contexts, they can easily be filtered by text.

Third, by attaching separate contexts to all Appellation Events and Relation Events, it is possible to directly compare different interpretations for the same text. For example, researcher A and B both annotate the same text independently from each other. The resulting graphs of that annotation process can then be compared to find differences or overlaps in the researchers interpretations of that text. Figure 13, scenario three shows a situation in which the graph for a text consists of three relationships. One relationship was created by both researchers A and B, while A and B also created an additional relationship independently.

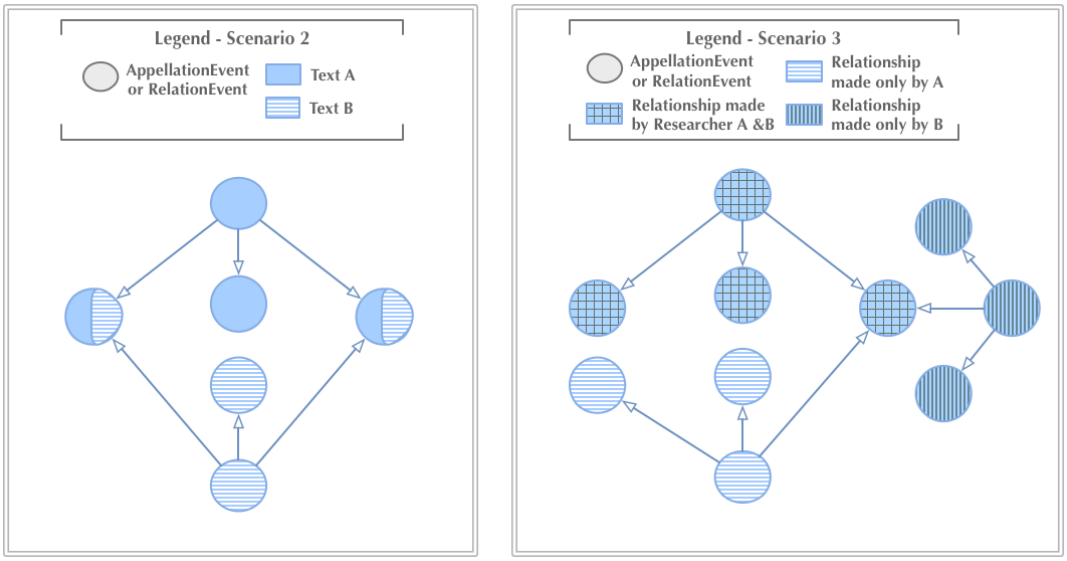

Figure 13: Filtering Graphs by Contexts and Comparing Graphs

4.2 Concepts

In 4.1.1 Appellation Events, Table 6, the property interpretation of Appellation Events is described as specifying the meaning of a term in a text. From this description two questions arise that are central to the Quadriga System.

a) How is the meaning of a term represented?

b) How do we know if two terms have the same meaning?

While a) is a rather technical question, the answer to b) has implications on the design of the whole system. In the Quadruple-based research system, the answer to a) is that meanings of terms are represented by entries in a global authority file service. For example, there is an entry for the person Albert Einstein, the scientific field physics, or the species of horses. Each entry is identified by an URI, which is used in the property interpretation of Appellation Events to specify the meaning of terms. However, the authority file does not try to capture an exact definition of its entries. All it specifies is that a certain concept exists, and some basic information about it, such as a type and a name. The types of concepts have a hierarchical structure with the most general type "Entity." For example, the entry for Albert Einstein is of type "Person" as it is probable that most people will agree that Albert Einstein existed, and that he was a human being. However, there are cases of concepts for which there is no agreement on of what type they are. In these cases the most specific but at the same time sufficiently general type is used. In the extreme case that there is no agreement at all, the type of a concept is "Entity."

Regarding question b), the underlying assumption of the system is that the specific meaning of a concept can best be defined as a set of concepts surrounding it and its relationships to the surrounding concepts [Kintsch 1998]. For example, the concept of Albert Einstein would be defined by relations such as when he was born, where he was born, his parents, and his work (see Figure 14[43]). Furthermore, the authority file does not impose

[43] For the ease of understanding, the relationships in this diagram are simplified compared to the relationships in Figures 12 or 13.

an ontology on its entries but a type hierarchy. This means that besides a subclass relationship between types, there are no predefined relations between types or entries.

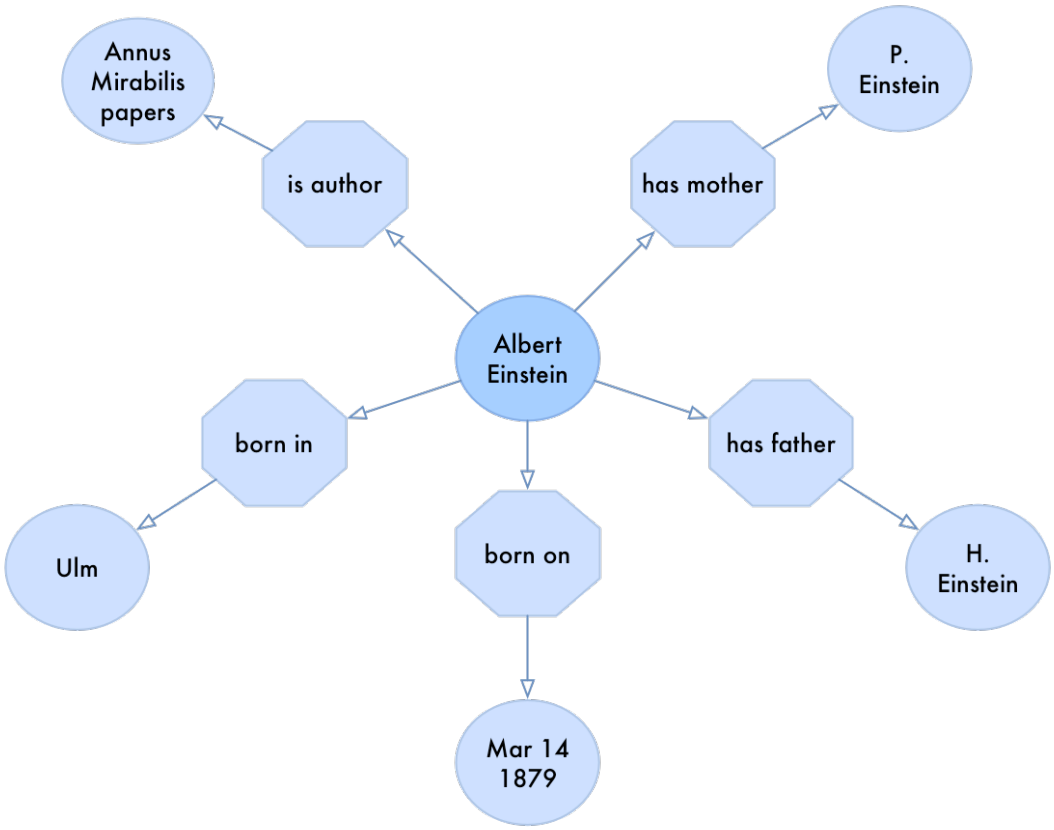

Figure 14: Concept Definition by Neighborhood

This approach of neither specifying any properties of concepts (besides very basic ones) nor any relationships between concepts has two advantages. First, the authority file can be used for a variety of fields. Scholars do not have to agree on any definition of a concept as long as they agree that the concept exists. Similarly, researchers can use the concepts in the authority file without having to worry about whether or not the authority file defines the relationships between concepts the way they would. Second, differences in definitions of concepts can be quantified and possibly detected in the data layer by

examining the relations between concepts. For example, consider two researchers each annotating several texts concerned with a specific concept. If the researchers have different understandings of the concept, it could be assumed that these differences would be detectable in the relationships the researchers create for the concept. If that is not the case, if all the relationships are the same or very similar, it might be possible that the definitions the researchers work with are not as different as assumed. Likewise, two scholars might annotate texts believing that they understand a concept the same way, but their relationships for that concept differ. This might indicate that the scholars' definitions of that concept are actually different.

4.2.1 Types of concepts

In section 3.2, I described ontologies and their structure. The authority file the Quadriga System is based on does not impose an ontology. However, to understand the nature of concepts as they are used here, it is useful to borrow some of the terminology of ontologies. Ontologies have classes and instances. Classes describe a group of similar elements (for example people). Instances represent specific members of a class. For example, a class "Person" would have an instance or member "Albert Einstein."

In the authority file service that is part of the Quadriga System, the differentiation between what are classes and instances is slightly different. The types used are the classes of the CIDOC CRM ontology. This event-based ontology was developed to describe cultural heritage data or museums data. The classes of the ontology describe concepts relevant to the museums world. For example, there are classes for specific events, such as the destruction, move, or acquisition of an object, or they describe the objects themselves (for example as biological object, or image). Some classes capture the components that are necessary for the

management of museums objects, such as who owns the rights over an object, or who is involved in a transaction such as acquisition of an object (for a full list of classes see [Crofts et al. 2011]). In the Quadriga System, the entries in the authority file can be seen as subclasses of these general CIDOC CRM classes. For example, the CIDOC CRM class E21 Person has subclasses such as Albert Einstein, Hans Spemann, or Johann Wolfgang von Goethe. Note that in many ontologies these concepts would be instances of a class Person rather than subclasses.

Another CIDOC CRM class that plays an important role in the system is E55 Type. "E55 Type is the CRM's interface to domain specific ontologies and thesauri" [Crofts et al. 2011, p. 23]. Since CIDOC CRM mainly defines classes to describe events or museums objects, this class is the connection point to add concepts such as species (grey whale or triops) or scientific phenomena and theories (natural selection or gravity). In the Quadriga System, every concept in the authority file that represents an idea, classification, theory, or any other kind of type, will have the CIDOC CRM type E55 Type. For example, the concept of sheep as a species will be of type E55 Type.

This separation of concepts into concepts subclassing a CIDOC CRM class (excluding E55 Type) and concepts subclassing E55 Type has some side effects that seem to be odd at first. The sheep Dolly, for example, would not be a subclass of sheep because that type does not exist. The type system of the Quadriga System is intended to classify concepts into very broad categories such as person, organism, or physical thing. Instead, Dolly would be of type E20 Biological Object, which is a class comprising "individual items of a material nature, which live, have lived or are natural products of or from living organisms" [Crofts et al. 2011, p. 11]. One reason for this method of assigning types is that classifying Dolly as a

sheep is an act of interpretation (even if it is a widely accepted interpretation). Another example might explain this method further. Consider the planet Pluto. Pluto was officially classified as a planet until 2006, when it was decided that it did not meet all the requirements for that classification. If Pluto would be a subclass of type planet, the concept itself would have had to be changed with the change in official classification. But if the concept Pluto would not be a subclass of the concept planet anymore, then older texts that still state that Pluto was a planet might create inconsistencies with the concepts hierarchy. If instead Pluto is classified as E18 Physical Thing and planet as E55 Type, the concepts are independent of each other in the context of the authority file and inconsistencies are prevented. Instead, concepts are linked by the annotations themselves that might state that Pluto is a planet or a dwarf planet.

4.2.2 Linking out

Section 3.1 discusses authority files such as the Virtual International Authority File (VIAF). Some authority files play an important role in the (digital) humanities and are used widely. However, usually these services are limited to a specific topic or subject. For example, VIAF contains mainly concepts such as people, places, or publications; GeoNames[44] is concerned with places. The Quadriga System requires an authority file that contains any kind of concept. Therefore, it uses its own authority file services.

Referring back to section 3.4, Berners-Lee lists a set of four principles that services in the web of linked data should follow. The fourth principle is that a service should "[i]nclude links to other URIs, so that they [people or software agents] can discover more

[44] See http://www.geonames.org/

things" [Berners-Lee 2006, First section]. The authority file used in the Quadriga System follows this fourth principle by providing two fields equals and similar to for each entry. By doing so, it links out to other authority files and controlled vocabularies. These two fields can contain URIs from other services. For example, entries for people usually link to the respective entry in VIAF. Similarly, some institution or publication entries reference VIAF. Geographic places often link to GeoNames, a database containing over eight million places such as cities, countries, or mountains all over the world. Each place in GeoNames has a URI that uniquely identifies it. Both fields, equals and similar to, can contain several URIs, GeoNames and VIAF for example, if a concept exists in both services.

The fields equals and similar to are similar fields as they both contain URIs to other authority services. However, while equals links to entries that represent exactly the same concept, similar to refers to concepts that are not necessarily the same concept or concepts that have slightly different interpretations attached to them. For example, for people it is often fairly easy to find an entry in VIAF that clearly describes the same person. However, concepts such as theories or ideas are often not contained in an authority file but are, for example, part of a Wikipedia entry. In these cases, the similar to field can be used to link to the overall entry.

There are two main advantages of linking to other resources in the way described above. First, the linked sources can be used to get information about the entries in the authority file of the Quadriga System. Most services offer an interface that a computer program can use to retrieve information about their entries. This means that a software application that is using the entries in the authority file of the Quadriga System could retrieve additional information from other sources about a specific concept to display it, or use it for

data analysis. For example, when displaying a node that links to the concept Albert Einstein, the application could use the link to VIAF to retrieve a list of publications by Einstein in order to display them to users.

Second, by linking the Quadriga System authority file to other services, it makes it possible to integrate existing external data sets that use other authority files or controlled vocabularies. The URIs used in those external data sets can be compared to the URIs stored in the two fields equals and similar to of the authority file entries used in the Quadriga System. This way existing data in the Quadriga System could be linked to existing external data sets.

4.3 Quadruples and Concepts

In the previous sections, I described how quadruples are structured using Appellation Events and Relation Events, and the role concepts play in the Quadriga System. In this section, I will examine the relationship between these two kinds of elements and how this relationship affects the Quadriga System.

Appellation Events are connected to concepts through their interpretation property. Relation Events are connected to concepts through the Appellation Events they link to. Concepts themselves do not relate to other concepts directly, as the authority file service that contains all concepts creates almost no connections between concepts. Therefore, the Quadriga System has two layers: the *events layer* that contains Relation Events and Appellation Events, and the *concepts layer*. From these two layers a third layer can be derived, the inferred layer, which can be understood as basis for a knowledge graph. This knowledge graph can be used to support the analysis of texts or to answer questions. In the inferred layer, concepts are linked based on the information in the events layer (see Figure 15).

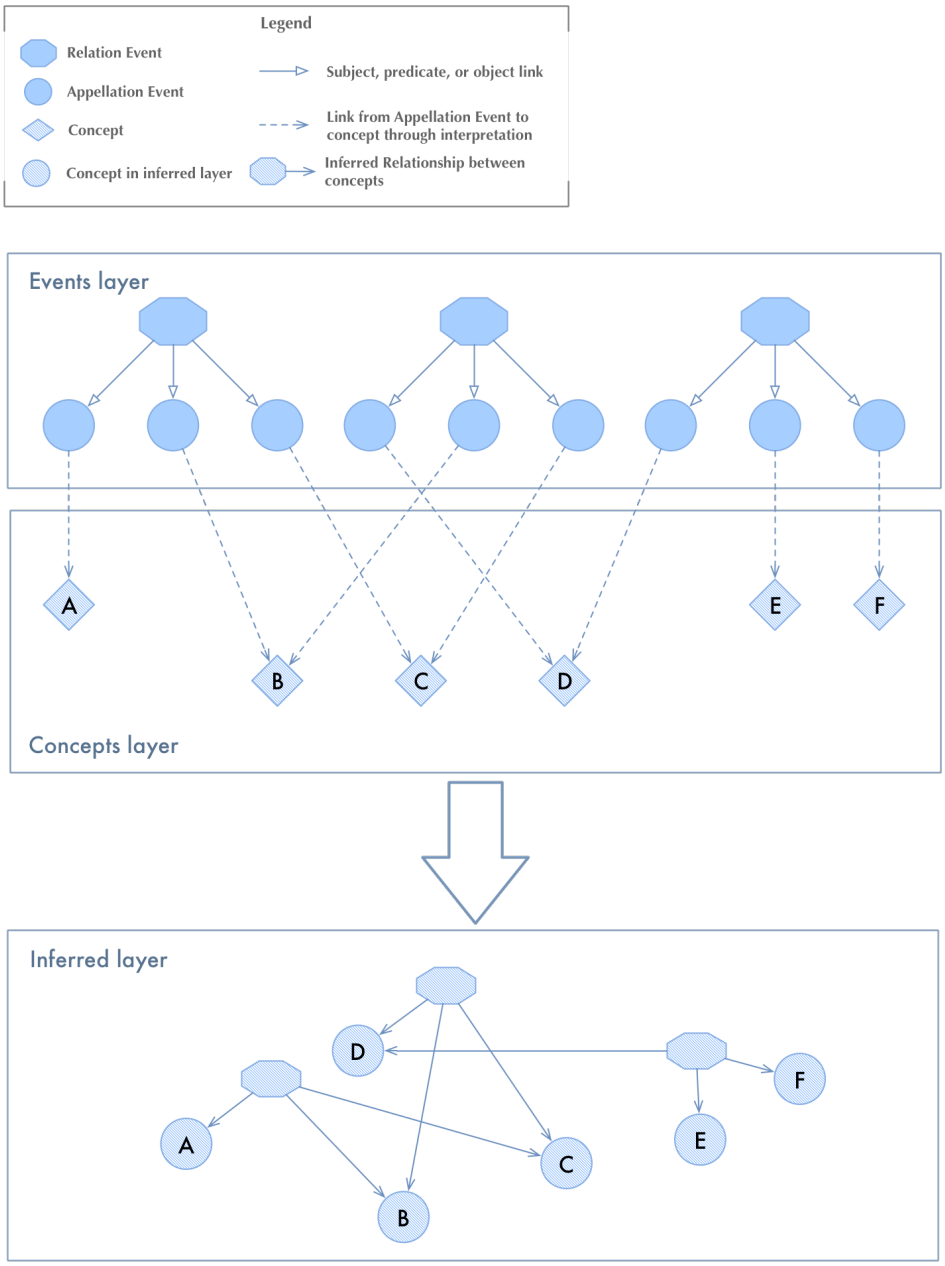

Figure 15: Layers of the Quadriga System

However, as seen in Figure 15, concepts in the inferred layer are still not directly linked, but linked through a relationship element (similar to a `Relation Event`) that specifies what concepts are subject, predicate, and object. The relationship element exists because, in the Quadriga System, quadruples can be nested. For the nesting of quadruples it is necessary that

the subject, predicate, object relationship, as a whole, can be referenced. This requires an element representing the relationship: the relationship element. If we wanted to truly link two concepts (the concepts that are used as subject and object), the concept used as predicate would have to be turned into a link between the other two concepts. In a graph theoretical way that would mean that the node that represents the predicate concept would be turned into an edge, and that edge would have all the properties of the predicate node (see Figure 16).

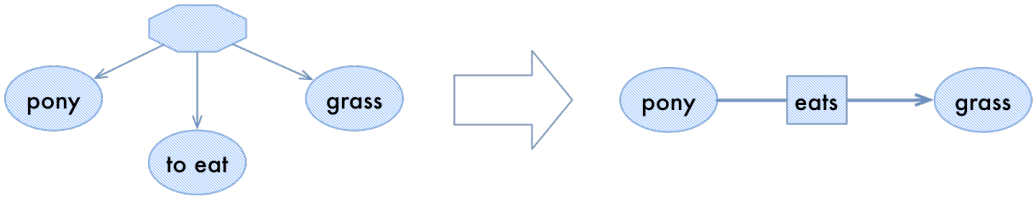

Figure 16: Transformation of a Relationship Node into an Edge between two Concepts

In the world of RDF, the described situation is a common problem. A statement such as the one on the left side of Figure 16 that uses an extra relationship element that defines subject, predicate, and object of a triple is called a *reified statement* [Powers 2003c]. Accordingly, the process of turning an edge into a node (going from right to left in the figure) is called *reification* [Powers 2003c]. In RDF, reification is a problematic technique and according to [Hellman 2009] not used very often. One reason for that is that RDF statements and queries become much more complex when using reification. Listing 7, for example, shows a simple query in pseudo code[45] for all subject, predicate, object triples,

[45] Pseudocode is code that describes a specific algorithm or procedure on a very high-level. The purpose of pseudocode is to demonstrate a principle or algorithm to a human reader. It is not intended to be parsed and executed by a computer.

which do not use reification, that have http://example.org/pony as subject. Listing 8, in contrast, shows the same query for reified triples.

Listing 7: Query without Reification (Pseudocode)

```
SELECT s, p, o WHERE
      { s = "http://example.org/pony", p, o }
```

Listing 8: Query over Reified Triples (Pseudocode)

```
SELECT s, p, o WHERE
      {
            r hasSubject s and s = "http://example.org/pony",
            r hasPredicate p,
            r hasObject o
       }
```

RDF and reification in RDF is relevant to the Quadriga System as RDF is a crucial technology for the Semantic Web. The networks and network analysis results created in the Quadriga System could potentially be highly valuable for the future development of Semantic Web agents. For example, a software agent could recognize that the information that the earth is flat rather than spherical is several hundred years old, and answer a query about the shape of the earth with an historical overview of this question. In the development of the Quadriga System, considerations regarding the Semantic Web, therefore, are critical.

Independent of the issues that arise from using reification in RDF, there are other problems associated with networks or graphs that are created out of reified triples or statements. The triple shown on the left in Figure 16 (the reified triple), when understood as a network, has two types of nodes: nodes that represent concepts and a node that represents the relationship element referring to the concepts. In the picture, these different kinds of nodes are represented by different node shapes (circles and octagon). A network with two different kinds of nodes is called a two-mode or bimodal network [Newman 2010]. While

there are several different measures and analysis techniques for networks with only one kind of node, techniques for analyzing two-mode networks are more complicated [De Nooy et al. 2011]. A lot of measures are well developed for networks with one mode, but to apply them to bimodal networks those first have to be transformed into one-mode networks [Opsahl 2013b,a, De Nooy et al. 2011].

In addition, one could argue that a network that consists of reified triples does not have two, but three types of nodes. Predicate nodes could be considered a different type than subject and object nodes. While concepts used as the subject and object of a triple are usually the entities being studied, predicates describe the relationships between those entities. Networks that consist of reified triples would therefore be three-mode (or multimodal) networks that are even more complex to analyze.

4.4 Visualizing Networks

An important functionality of the Quadriga System is to visualize the networks created by researchers. So far, the Quadriga System provides basic visualization of networks, which needs to be extended as analysis functionalities are added to the system. This section describes the basic assumptions regarding visualization of networks and the rules applied to networks when visualized.

In the Quadriga System, the only persistent networks are the ones in the events layer consisting of Relation Events and Appellation Events. Networks, as shown in the inferred layer, are dynamically generated as needed. For example, for visualization purposes. However, the reified triples shown in the inferred layer in Figure 15 are transformed before they are being visualized as shown in Figure 17.

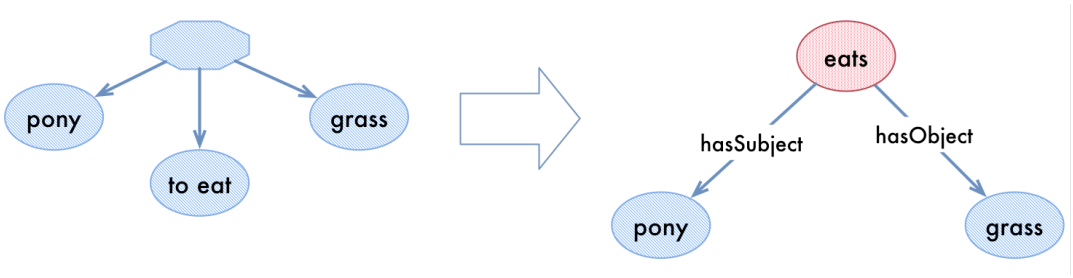

Figure 17: Transformation of a Reified Triple in the Quadriga System

The relationship element is removed, and there are two outgoing edges added to the predicate node, a hasSubject and a hasObject edge, that connect to the subject and object nodes. The predicate node can be assigned a different color or shape than the subject and object nodes to make the triple easily readable. To refer to a triple from another triple by using it as subject or object (to nest triples), the subject or object edge can point to the predicate of the "nested" triple. Figure 18 shows how the statement "John says that ponies eat grass" could be represented using nested triples.

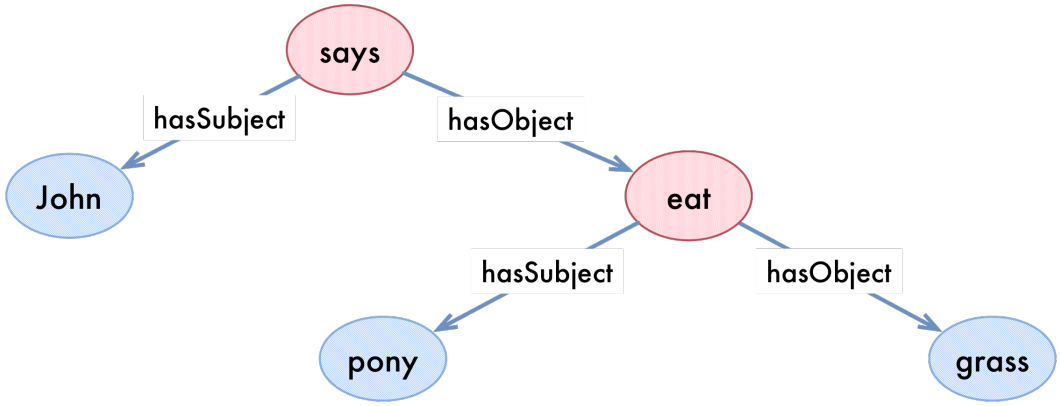

Figure 18: Nested Triple - Use a Triple as Subject or Object of Another Triple

However, the following problem arises from the form of visualization described above. If there are two or more triples that use the same concept as predicate, this kind of presenting triples become ambiguous. The box in Figure 19 demonstrates the problem that arises when two triples use the same concept as predicate: it is not clear anymore which hasSubject and hasObject edges belong to the same triple. The triple could either state that ponies eat grass or that they eat meat.

While the information to disambiguate such situations can be kept in the data model on which the visualization is based[46], it makes it difficult for a human user to interpret and analyze such networks. The visual interpretation of a human user becomes even more difficult as networks using the described triple representation can become very cluttered and almost impossible to read (see for example Figure 19).

To solve this problem, in the Quadriga System subject and object nodes are "reused"—they can be linked from several different triples. However, predicate nodes are unique to a triple and have exactly two outgoing edges: one pointing to the subject and the other to the object of a triple. Predicate nodes can have multiple incoming edges, though. For example, the statement that ponies eat grass can be made by two different people resulting in two triples both pointing to the statement "ponies eat grass" as object. Figure 20 shows how the network shown in Figure 19 would be represented in the Quadriga System.

[46] This means that the computer still knows which subject and object belong together and can use this information for calculations. However, when creating a visualization those information is not presented.

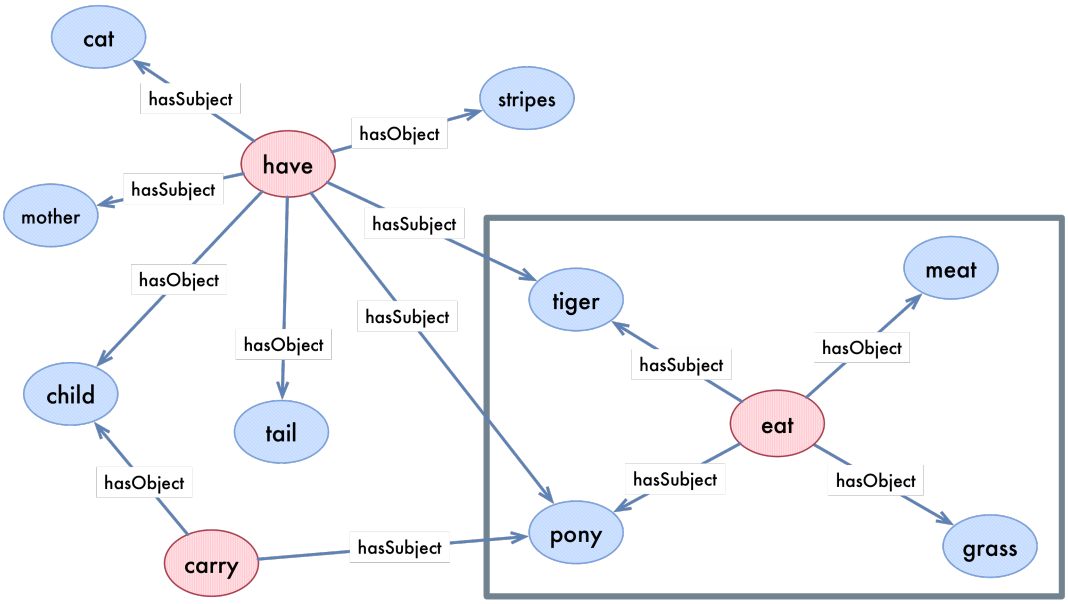

Figure 19: Cluttered Ambiguous Visualization of Subject, Predicate, Object Triples

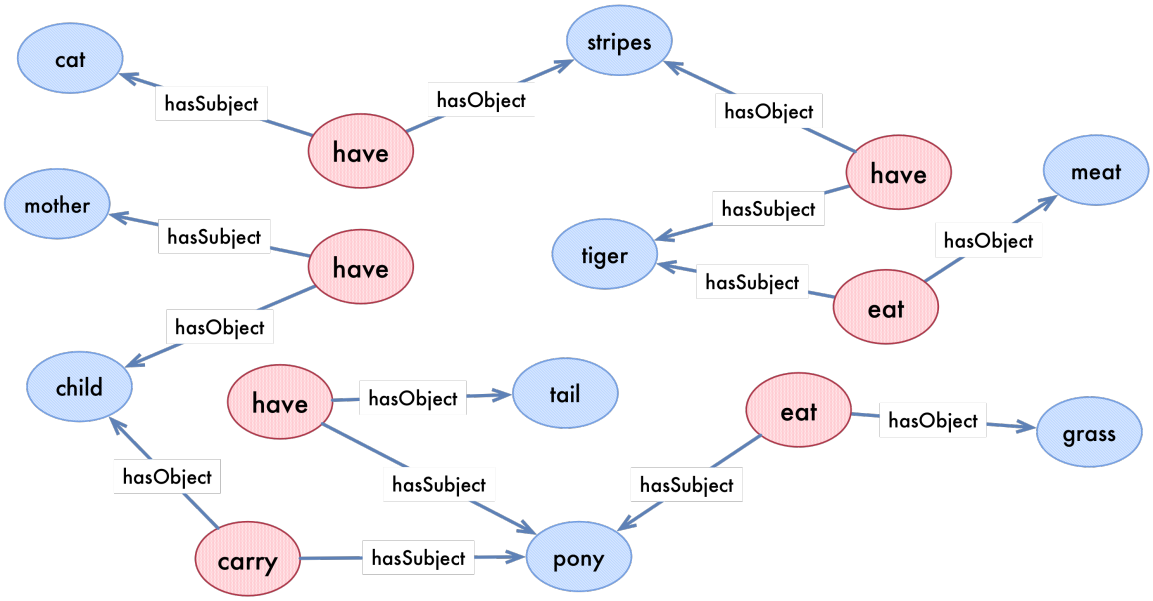

Figure 20: Network from Figure 19 as Visualized in the Quadriga System

This kind of visualization of triples assumes that subject and object nodes are of "greater interest" than predicate nodes. Being of greater interest here means that measures such as the degree of nodes (how many edges does a node have [Caldarelli and Catanzaro 2012a]) or centrality (how important is a node in a network [Caldarelli and Catanzaro 2012b]) make more sense to be applied to subject or object nodes rather than predicate nodes. For example, in the network shown in Figure 19, the node with the highest degree is have. However, this does not say much about the network or the text that the network was created for. In the same network, when represented in the Quadriga System (Figure 20), the node pony has the highest degree, which tells a human reader that there are three statements that were made about ponies. This could mean that the text for which this network was created might talk about ponies or animals in general.

Focusing on subject and objects rather than predicates is a compromise between representing exactly the information stored in the system and visualizing the information in a useful way. For questions that are concerned with the interaction between two concepts rather than the concepts themselves, this kind of visualization might not be ideal. For example, if a researcher is interested in what kind of food is eaten by animals, a network visualization that has only one node for the concept eat might be more helpful. These issues need to be kept in mind and addressed for future development of the Quadriga System.

4.5 Ontologies Bottom-Up and Top-Down

In section 2.3 (Semantic Web) and section 3.2 (Ontologies in Computer Science), I discussed ontologies and the problems they can pose to the history of science. As mentioned above, the Quadriga System imposes, only to a very limited extent, a predefined structure on the concepts used in the system. There is no global ontology that specifies how concepts relate

to each other. However, ontologies are a central part of the design of the Quadriga System. This section describes how ontologies fit into the system's design.

As pointed out in [Nagypal et al. 2005], in the field of history (and therefore also history of science), it is especially challenging to create ontologies due to factors such as the change of definitions of concepts over time, or uncertainty of information. Therefore, the Quadriga System does not try to define an overarching ontology, but supports the creation as well as validation of ontologies. In the *bottom-up* approach described here, ontologies can be created out of the annotations (Appellation Events and Relation Events) stored in the Quadriga System. The *top-down* approach lets a user of the system validate an ontology against the stored annotations. However, while the technical prerequisites for implementing the two approaches are provided, they are currently not yet realized.

4.5.1 Creation by Annotation

In the bottom-up approach, the annotations that are stored in the Quadriga System can be used to create an ontology. Appellation Events refer to concepts that have very basic types assigned (for example `E21 Person`). If a researcher wants to specify a more concrete type of a concept, he has to do that by creating a Relation Event using the predicate `be` in the sense of "is narrowed by"[47]. This predicate represents a relationship between two sets of elements, one of which is a sub-group of the other[48]. For example, `<animal - be (is narrowed by) - bird>` states that all the elements in the group bird are also in the group animals. In extreme cases there is just one element in the subgroup. For instance, the following triple states that Donald Duck is a duck: `<duck - be (is narrowed by) - Donald Duck>`).

---

[47] Instead of "is narrowed by," this relationship could also be called "broadens."

[48] "Narrower" relationships between terms in combination with "broader" relationships are typically used in thesauri to specify term hierarchies [Colomb 2002].

A piece of software could go through all Relation Events stored in the Quadriga System, extract those Relation Events that follow the above structure and create subclass relationships between the subject and object of those Relation Events (see Figure 21). To identify what is a class and what is an instance, the types of the concepts can be used. Concepts of type `E55 Type` would be classes; concepts of other types such as `E21 Person` or `E71 Man-Made Thing` would be instances. In cases where this process is not suitable, additional rules could be developed; such as that the lowest elements in the hierarchy are instances.

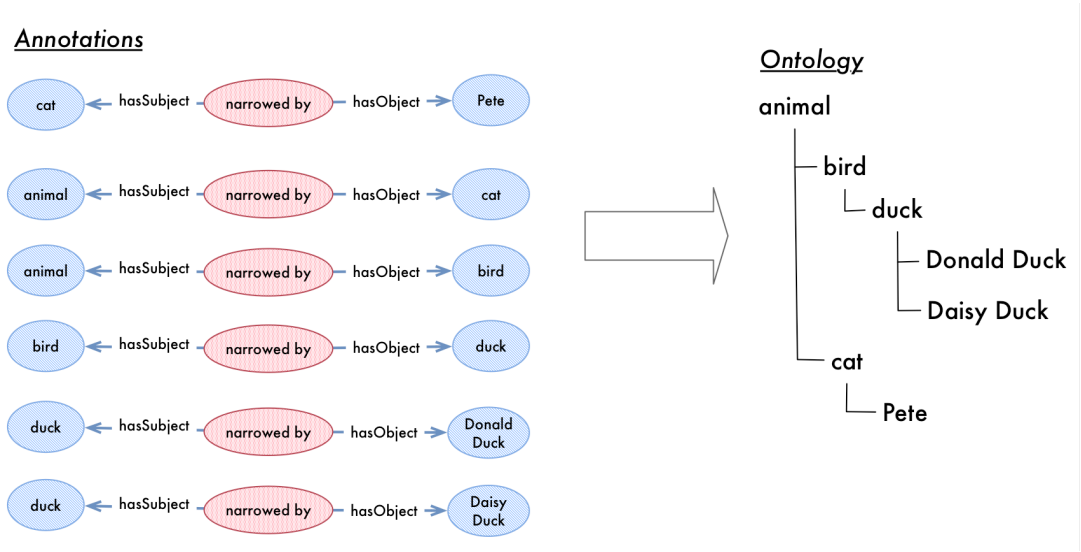

Figure 21: Creating an Ontology out of Annotations

However, creating an ontology out of all stored Relation Events would likely cause the same problem that the Quadriga System tries to avoid: different or changing interpretations, and classifications might not fit into one subclass hierarchy. The problem could be solved by restricting the annotation-base used to create an ontology. For instance, only annotations created for texts that were published between 1800 and 1820 might be

considered, or only texts written by a specific author. This way it would be possible to create several ontologies (for example one for each ten year period) and compare those to investigate how the classification of certain entities changed over time.

To incorporate relationships between classes of an ontology, other Relation Events (besides be (is narrowed by)) could be considered. For example, if there is a Relation Event that states that physicists run experiments, a software application could create a relationship run between the two classes physicist and experiment. This process, however, would not be as straightforward as creating subclass relationships and would need more analysis. Questions like: "When can a relationship on the instance level be generalized to the class level?" still need to be answered. For instance, if there are statements specifying that Albert Einstein wrote the Annus mirabilis papers, at what point (how many statements are needed) can the relationship be generalized to claim that physicists write papers?

4.5.2 Validating Ontologies

The top-down approach can be understood as being the opposite of the bottom-up approach. Instead of starting with annotations and creating an ontology out of them, the top-down approach starts with an ontology and then tries to fit existing annotations onto that ontology. For example, the annotations stored in the Quadriga System could be used to validate existing ontologies by applying the ontologies to the stored annotations. If the annotations are consistent with the ontology, the ontology would be valid. In cases of annotations that are not consistent with an ontology, this could point to either difficulties with the ontology, texts that differ from the assumptions underlying the ontology, or annotators using a different conceptualization than the researcher who created the ontology.

Similarly to the bottom-up approach, in the top-down approach Relation Events could be analyzed to find hierarchical relationships. Those relationships could then be compared to the given ontology. For example, the ontology to be validated could state that any member of the class bird cannot be also a member of the class cat. However, in the annotations stored in the Quadriga System there might be two Relation Events, one stating that Pete is a cat and one that Pete is a duck. Translated into an ontology structure, that would mean that Pete is both, an instance of the class duck, which is a subclass of bird, as well as of cat. Given those annotations, the ontology would not be valid.

If an ontology cannot be validated against a set of annotations, a researcher can examine the reason for a failed validation for an explanation. As Appellation Events and Relation Events contain references to the positions in the texts for which they were created, the cause for a failed validation can be traced back to the source of a Relation Event or Appellation Event. This enables researchers to study the causes of conflicts or inconsistencies of annotations and ontologies, and to possibly resolve those.

4.6 System architecture

The Quadriga System consists of several independent components that interact with each other (see Figure 22). Each component has specific responsibilities. A user might directly interact with all components or with only a few depending on his role in a project. This section will illustrate how the components work together. In section 4.7, I will describe specific implementation details.

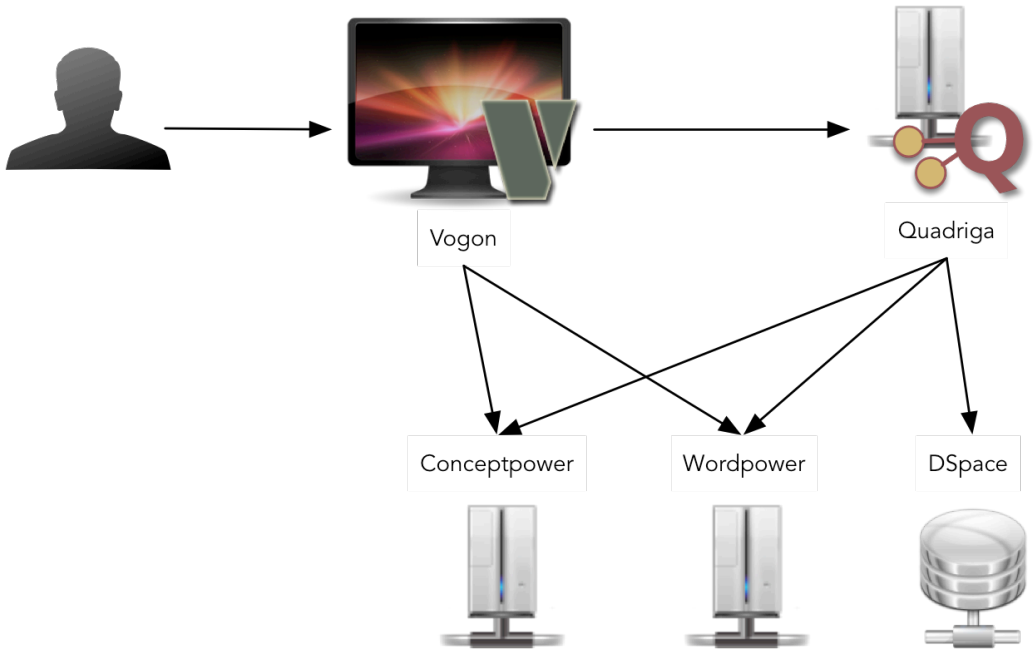

Figure 22: Outline of Quadriga System Architecture

The component that users will likely interact with the most is *Vogon*. Vogon is a desktop application that enables users to annotate texts with Appellation Events and Relation Events. This can be done with a text-based editor, in which users highlight the terms that they want to annotate, or by using a graphical editor that lets users build a graph diagrammatically and then connect each node in the graph to the text. In 4.7.1 (Vogon), I will describe the annotation process in more detail.

Appellation Events and Relation Events together form graphs. When a user has finished annotating a text, those graphs can be submitted to *Quadriga*, a Quadruple repository. Quadriga is the central component of the Quadriga System. It is a web application that provides functionality to review, annotate, store, and publish graphs

consisting of Quadruples. In the context of Quadriga, a graph consists of all the annotations that were created for a text by one user that are submitted together at the same time. For example, if annotator A creates annotations for text X and submits those annotations to Quadriga, Quadriga contains one graph for text X. If now annotator B annotates text X and submits his annotations to Quadriga, Quadriga holds two graphs for text X. If annotator A annotates text X again and submits the annotations, Quadriga contains three graphs for text X. I will call those graphs *text graphs* or *text networks*. Each text graph can connect to other text graphs by referring to the same concepts. Hence, all text graphs stored in Quadriga form themselves a graph (see Figure 23). I will call this graph, which spans multiple text networks, the *overall graph*. Referring back to section 4.3 Quadruples and Concepts, Figure 15, the overall graph is the graph that spans the events layer and the concepts layer.

Quadriga follows a project-centric approach. This means that the most general elements in Quadriga are so-called *projects*. A project contains *workspaces*, which contain texts. Workspaces can be used to group together texts in a project. For example, if there are three researchers working on specific texts in a project, each researcher can create their own workspace and add only the texts they annotate to their workspace. A workspace can be *checked out* by a researcher using Vogon. This means that all the texts contained in a workspace are downloaded to the computer that runs Vogon for a researcher to annotate.

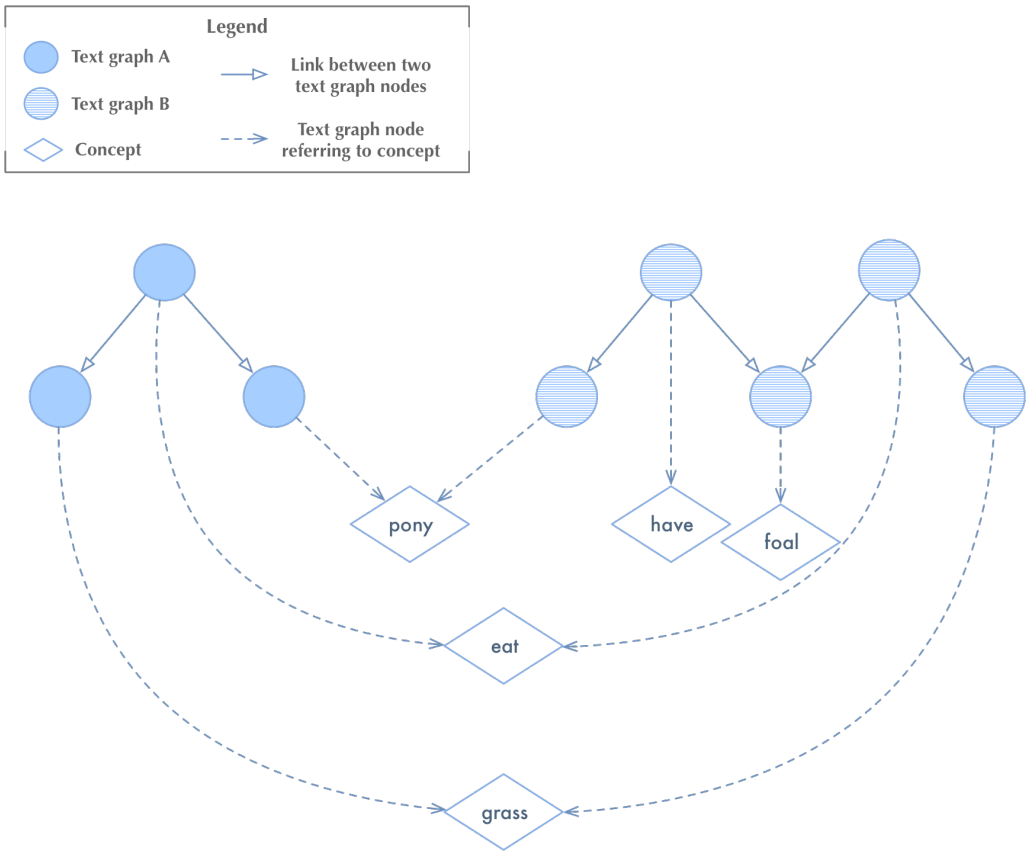

Figure 23: Overall graph — Two Text Graphs (or Text Networks) are Connected by Referring to the Same Concepts

Each project has *collaborators* and *editors*. Collaborators are researchers annotating texts. Editors, however, review submitted text networks, make annotations to them, and either reject them if they find that modifications to the networks are needed or approve them for publishing. If an editor rejects a text network, the annotator can download the annotations the editor made for a network and modify the network as needed before he resubmits it to Quadriga. If an editor approves a text network, the network will be published,

which means that it is added to the repository permanently and that it becomes part of the overall graph.

Texts are fundamental to the Quadriga System: quadruples are created for texts, Appellation Events link to positions in texts, and Quadriga manages graphs by associating them with specific texts. To use the Quadriga System to its full potential, documents that are being annotated using Vogon should be available to the whole system. This would allow visualization websites of annotation graphs to display the part of a text for which an annotation was created. In the Quadriga System, texts are made available through a DSpace repository[49], called *HPS Repository*. It can be accessed through http://hpsrepository.asu.edu/.

In a DSpace repository, every so-called "item" is assigned a handle[50] [Diggory and Luyten 2013]. For example, an item represents a paper, a presentation, or a book. Each item can contain one or more "bitstreams." A bitstream is an electronic version of the item, like a plain text file containing the text of a document, or a PDF version of a document. A handle is a unique identifier for an item in DSpace [Diggory and Luyten 2013]. It can be used to retrieve an item's metadata and its bitstreams. A good example of this is the Embryo Project article "Carl Richard Moore (1892-1955)" by Mary Drago, which is stored in the HPS Repository, and is represented by the handle http://hdl.handle.net/10776/7558.

The last two components in the Quadriga System are an online authority file service called *Conceptpower*, and an online dictionary service called *Wordpower*. Both services are web applications. Quadriga as well as Vogon interacts with these services though a web API

---

[49] See http://www.dspace.org/

[50] There are other elements in DSpace that are assigned handles as well; see [Diggory and Luyten 2013] for further information on that topic.

(Application Programming Interface)[51]. In contrast, human users interact with Conceptpower and Wordpower through a website using a web browser.

Conceptpower is the authority file system used in the Quadriga System. It contains all concepts and their properties as described in section 4.2. Each entry in Conceptpower represents a concept and is identified by a URI. Given such a URI, an application can retrieve a concept's properties, such as its type or the contents of the equals field. If a concept is missing in Conceptpower, a user can create a new entry in Conceptpower for the missing concept.

Wordpower has many similarities with Conceptpower. As in Conceptpower, every entry in Wordpower is identified by a URI and given the URI other software applications can request information about a Wordpower entry. The biggest difference between the two services is that in Conceptpower each entry represents a specific meaning of a term. Even if two terms are the same, if they have different meanings there will be different entries in Conceptpower. In contrast, with Wordpower there is only one entry for a term and that entry specifies the normalized or "correct" spelling of a word[52]. For example, Conceptpower contains five different definitions for the term "pony" such as "a range horse of the western United States" or "a small glass adequate to hold a single swallow of whiskey." In contrast, Wordpower has only one entry for the term "pony" that combines all five entries in Conceptpower. If there exist several meanings for a term, all of them are covered by one Wordpower entry. If an annotator creates an Appellation Event for the term "ponies" in a

---

[51] An API is "a way for two computer applications to talk to each other over a network (predominantly the Internet) using a common language that they both understand" [Jacobson et al. 2011, p. 5].

[52] The normalization of a term could be singular for plural nouns, present tense for verbs, or simply the correct spelling of a word.

text, it would point to the normalized spelling of the term in Wordpower and the meaning of the term in Conceptpower.

A typical annotation workflow using the Quadriga System looks like the following: A researcher starts by creating a workspace in an either existing or new project in Quadriga. He then adds the texts he plans to annotate to that workspace. Next, he opens Vogon and checks out the workspace from Quadriga. All the texts he added to his workspace are being downloaded to his computer. Using Vogon, the user creates Appellation Events and links those using Relation Events. During that step, he queries Conceptpower and Wordpower for the terms and concepts he uses in his annotations. He also creates new entries in these two services if terms or concepts are missing. Once the researcher has finished annotating a text, he submits the graph consisting of Appellation Events and Relation Events to Quadriga where his graphs are checked by editors and either rejected or approved and published.

4.7 Implementation Details

The software applications described in this section are central components of the Quadriga System. I developed prototypes for each of these components in the process of my dissertation studies. In the beginning of 2013, the Digital Innovation Group[53] consisting of computer science Master's students and students from the Biology and Society program was established. The development of all the software components of the Quadriga System was continued by the Digital Innovation Group under my direction. All software developed by

[53] For more information see http://devo-evo.lab.asu.edu/diging.

the Digital Innovation Group is open-source software and published on public repositories such as SourceForge[54] or GitHub[55].

4.7.1 Vogon

Vogon is a desktop application to annotate texts with Appellation Events and Relation Events. The networks a user builds by creating these kinds of annotations can be uploaded to Quadriga for visualization, exploration, and analysis. Vogon uses concepts stored in Conceptpower and terms provided by Wordpower for the creation of Appellation Events. This section will describe the general structure and implementation of Vogon. Screenshots and code examples can be found in Appendix B.1.1. The current version of Vogon along with its code and documentation can be found at http://gobtan.sourceforge.net/.

Vogon is being developed in Java using the Eclipse Rich Client Platform, a framework that supports the development of desktop applications and rich clients by offering a number of plugins that provide common functionalities such as file management, text editing, or version control. To store the annotations that a user creates, Vogon uses the Eclipse Modeling Framework (EMF)[56]. EMF stores data in XML format and provides functionality for easy access and modification of the data.

Vogon provides two different kinds of editors to create Appellation Events and Relation Events: a text-based editor and a graphical editor. In the text-based editor (shown in Figure 24) a user can select specific words in a text and annotate them with Appellation Events. Terms that have an Appellation Event attached to them are highlighted in the text.

---

[54] http://sourceforge.net/
[55] http://github.com/
[56] Information about EMF can be found at https://www.eclipse.org/modeling/emf/.

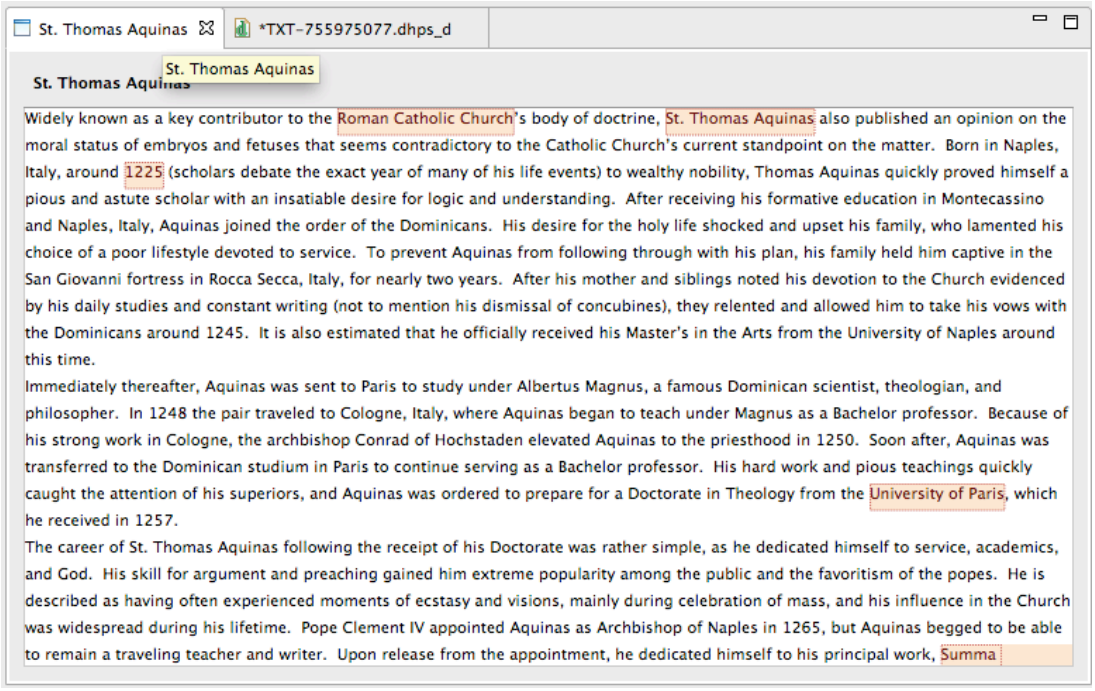

Figure 24: Vogon's Text-based Editor to Annotate Texts

The graphical editor (shown in Figure 25) lets a user build a graph by creating a diagram. Each node in the diagram can be connected to terms in a text. Graphs built with the graphical editor are visualized similarly to the visualization described in section 4.3. Predicates (displayed as blue boxes with green circles inside) link subjects and objects (represented as yellow circles) and are used as references in nested triples.

To simplify the creation of Appellation Events, Vogon can connect to Conceptpower and Wordpower. A user can search for concepts in Conceptpower or terms in Wordpower, add them to lists in Vogon, and then use the retrieved concepts and terms when creating Appellation Events. Vogon automatically assigns the URI of a concept or term to the appropriate property of an Appellation Event. This ensures that every Appellation Event that refers to a specific concept (such as Albert Einstein) uses the same concept in Conceptpower.

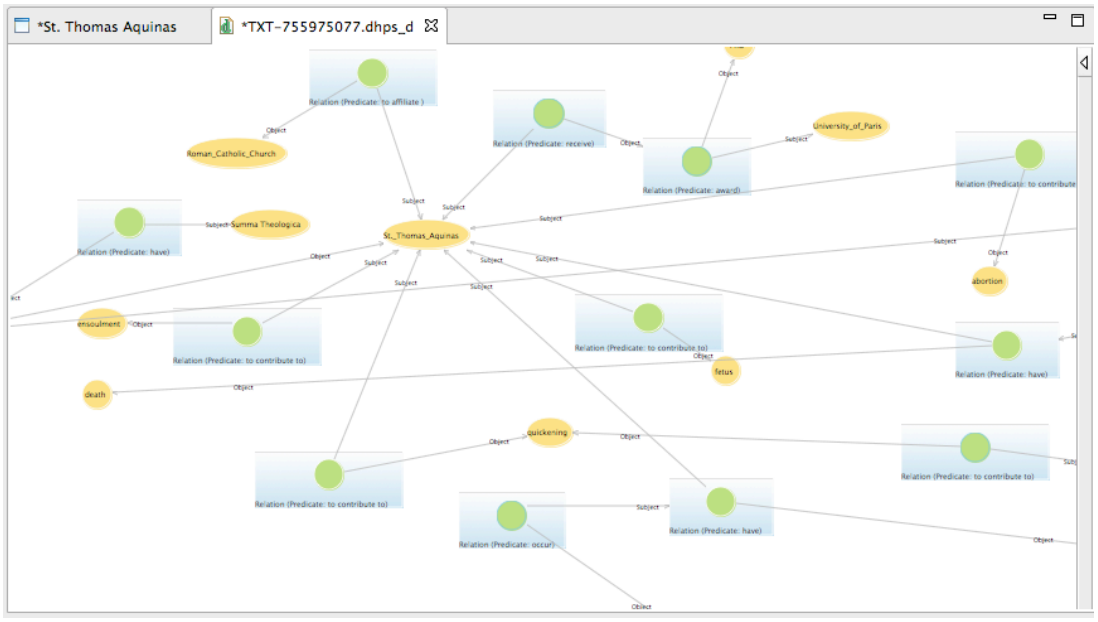

Figure 25: Vogon's Graphical Editor to Annotate Texts

While the Quadriga System is built on the idea that there is no predefined vocabulary that limits the possible relationships between two concepts (if a predicate is missing it can simply be added to Conceptpower to be used in Appellation Events), it became clear during the development of the Quadriga System that there are certain types of statements that occur repeatedly in texts or projects. For example, if a project is interested in what kind of relationships there are between people of a certain group, there are usually a limited number of possible statement structures. For example, two people can co-author a paper, collaborate on a project, or simply share an office. In these cases it is useful to define a set of templates that define these structures and that can be used by all users annotating texts for the project. In Vogon these templates are called *Standard Graphs*.

Standard Graphs are files that contain a graph in GraphML format[57], which is a form of XML. Each node of a Standard Graph can have a URI attached to it that links a specific concept. When a user uses a Standard Graph to create a new statement, Vogon creates the necessary Appellation Events and Relation Events. For nodes in the Standard Graph that have a concept attached to them, Vogon assigns the appropriate concept to the Appellation Event. Nodes without concepts need to be assigned a concept by the user.[58]

A feature that uses Standard Graphs and that gives Vogon many potential applications is the so-called *graph mapping*. This feature allows users to specify mappings for Standard Graphs that transform parts of a network created with Vogon, the parts that match a Standard Graph, into different kinds of networks.

For example, a Standard Graph could describe statements asserting that a specific person co-authored a paper with another person. The Standard Graph for such statements could look like the one shown in Figure 26. The italic font of node labels indicates that those nodes are placeholders for nodes that link to concepts with the specified type Person.

Figure 26: A Simple Standard Graph defining Co-Author Relationships between People

[57] See http://graphml.graphdrawing.org/ for further information
[58] Section B.1.1 contains an example Standard Graph as it could be represented in a diagram tool and in GraphML.

If a project used the Standard Graph shown above to encode co-authorship relationships, the graph-mapping feature could be used to find all the statements that have the specified form, and to transform them into "simpler" graphs in which, for example, two nodes are connected by an edge, when there is a co-author relationship between these two people. Such a transformed network, for instance, could be analyzed using social network analysis methods.

The advantage of using Vogon rather than another network software to create networks of co-authors would be that the text positions of the statement that assert a co-authorship would be available to be used when visualizing the transformed network. Also, several users can work on the same network: for example, if a number of texts are used for the basis of the network. The users (or annotators) only have to split up the texts and annotate each text individually. Conceptpower ensures that the same concepts are referenced and Standard Graphs guarantee that the same network structures are created. Using the mapping feature, the annotations created for all texts can then be exported and combined in one large network for further analysis. In regard to the Quadriga System, Vogon ensures that annotations are created in the correct format to add them to Quadriga. Annotators can work locally, focusing on the texts they are interested in, while also being able to contribute to a larger dataset consisting of annotations from a number of different projects by simply uploading annotated texts to Quadriga. Two projects that use Vogon and the Quadriga System are described in more detail in the next chapter.

4.7.2 Conceptpower

Conceptpower is a web application that provides online access to an authority file. The entries of the authority file can be searched, modified, added, and deleted through a website.

Other software applications such as Vogon or Quadriga can connect to Conceptpower using its web API. This section describes specific implementation details and the general structure of Conceptpower. In Appendix B.1.2, screenshots and code examples can be found. The current version of Conceptpower as well as its code can be found at http://conceptpower.sourceforge.net/.

Conceptpower is written in Java. It was originally developed using Java-Server Faces (JSF), but was later ported to the Spring Framework[59]. Conceptpower is based on WordNet 3.0, a database containing nouns, verbs, adjectives, and adverbs of the English language [Princeton University 2013]. For each meaning of a term, WordNet contains a separate entry. In addition, it specifies synonyms of terms that create so-called "synsets," "sets of cognitive synonyms" [Princeton University 2013]. WordNet is freely available and can be downloaded at http://wordnet.princeton.edu/wordnet/.

Conceptpower provides a web front end (a website) and a web API to search the WordNet database. For each entry, Conceptpower creates a unique URI. However, WordNet does not contain all possible concepts in the world. Especially lesser known people or institutions, or very specific technical terms are often missing. Conceptpower, therefore, can be extended by adding new concepts. Excluding concepts provided by WordNet, the Conceptpower instance used in the Quadriga System contains about 4500 entries as of March 2014. About 2500 of those entries represent a person and around 450 concepts are of type E40 Legal Body, which is used for concepts such as institutions.

A concept in Conceptpower has the following properties listed in Table 8.

[59] The Spring Framework is an open-source Java framework that provides a developer with comprehensive functionalities such as database support, web services support, and other features.

Table 8: Properties of Concepts in Conceptpower

PROPERTY	DESCRIPTION
Term	The "name" of a concept (for example "cat" or "Donald Duck").
URI	The URI that identified a concept.
Id	An internal identifier for a concept that uniquely identifies an entry in Conceptpower.
Wordnet Id	If a concept is contained in Wordnet, this is its id in Wordnet.
POS	Part of speech for a concept (noun, verb, adjective, adverb, or other).
Description	Description of a concept. The description is used by human users to decide if a concept represents the desired "thing" in the world.
Concept List	List of concepts that a concept belongs to (each concept can only be in one list).
Type	Type of a concept. The type is a very broad categorization of a concept. It should be general enough so that most users agree on it but specific enough to be meaningful.
Synonyms	Synonyms of a term representing a concept. This is mainly used for concepts contained in WordNet.
Equals To	URIs referring to entries in other authority files such as VIAF that represent the same concept.
Similar To	URIs referring to entries in other authority files that are similar to a concept. For example, if a concept is a specific tool, the similar to field could refer to a Wikipedia article describing the usage of that tool.
Creator	Information about who created a concept and when.
Modified	Information about who modified a concept and when.

Concept Lists

Each concept in Conceptpower belongs to exactly one concept list. However, these concept lists are for organization purposes only. They are employed by users to find a concept, or to build their own lists when adding new concepts. For instance, concept lists could group concepts that represent people or institutions, or group them according to a project's topic. All concepts that are contained in WordNet are put into a concept list called "WordNet" by default.

Concept Types

In Conceptpower, each concept has a type. Types can be defined for each Conceptpower installation. In the Quadriga System, types correspond to the classes used in CIDOC CRM (see section 2.3). The most general type is called `E1 CRM Entity`. Every other type in CIDOC CRM and Conceptpower is a subtype of this class. The only type present in the Quadriga System's Conceptpower that is not part of the CIDOC CRM type system is a type called `Predicate`, which is used for all concepts that represent predicates. For example, "to end something" as an activity is of type `Predicate`. Like concepts, every type in Conceptpower has a URI that can be used to uniquely identify a type. As of March 2014, Conceptpower contains 87 types.

WordNet Wrapper

As shown in Table 8, concepts in Conceptpower have properties such as equals to or type. WordNet entries do not have all of these properties. For example, there is no type specified for entries in WordNet. Conceptpower therefore uses an approach called *WordNet Wrapper*. A WordNet Wrapper takes a WordNet entry and adds the missing properties to that entry. For example, there is a noun entry for Albert Einstein in WordNet with the WordNet id

WID-10954498-N-??-einstein. The WordNet Wrapper that was created for this entry adds a type to the entry (E21 Person), a second id (CON1c268257-3f0e-4059-b4bb-a394ae2ce2a8), as well as a link to VIAF (http://viaf.org/viaf/75121530). Like adding additional concepts not present in WordNet, WordNet Wrappers need to be created by hand. Once created, they show up in search results of Conceptpower like any other concept. A WordNet Wrapper can be recognized by having two different values in the properties id and WordNet id. A WordNet entry that is not wrapped has the same value in both of these fields. Additional concepts have only an id but no WordNet id.

Conceptpower's Web API

Conceptpower provides a web API that can be used by other applications to query Conceptpower. Simply speaking, an application calls Conceptpower by accessing a specific URL that contains information about the application's request. Conceptpower extracts the information and returns the requested data in XML form. For example, an application such as Vogon could send a request to the URL:

http://chps.asu.edu/conceptpower/rest/Concept?id=CON1c268257-3f0e -4059-b4bb-a394ae2ce2a8

The URL consists of two parts, a path and a parameter, separated by a question mark character. The path (http://chps.asu.edu/conceptpower/rest/Concept) tells Conceptpower what kind of information is requested. In the given example, information about a specific concept is requested. The parameter tells Conceptpower what concept is requested (the one identified by the id CON1c268257-3f0e-4059-b4bb-a394ae2ce2a8). Conceptpower retrieves

the requested data and returns it to the requesting application in XML format (an example response can be found in Appendix B.1.2).[60]

Linked (Open) Conceptpower

As detailed in section 3.4, according to Berners-Lee, services that want to connect to the web of data should provide HTTP URIs for the "things" they publish, provide their content using standards such as RDF, and refer to other services by including other URIs [Berners-Lee 2006]. Conceptpower follows rules one and two by identifying each concept and type using URIs (rule one) that are HTTP URIs (rule two). For example, Albert Einstein is identified by http://www.digitalhps.org/concepts/CON1c268257-3f0e-4059-b4bb-a394ae2ce2a8. Conceptpower also follows rule four by providing the two concept properties similar to and equal to that link to other authority files such as VIAF. The only guideline that Conceptpower does not follow is providing its content using standards such as RDF or SPARQL[61]. To connect Conceptpower to the Linked Data network, this is a crucial step that needs to be addressed in Conceptpower's future development.

4.7.3 Wordpower

Wordpower is a web application that provides access to an online dictionary that provides guidelines regarding the spelling of words. Like Conceptpower, Wordpower's entries can be searched, added, modified, and deleted through a website. Other applications can query Wordpower using its web API. This section describes Wordpower's implementation and functionality. Screenshots and code snippets can be found in Appendix B.1.3. The current

---

[60] This is just one example of a so-called *web service* (a service that can be accessed through a web API). There are several standards and technologies for creating web APIs.

[61] sparql is a query language to query and manipulate RDF data.

version of Wordpower as well as its source code can be found at http://wordpower.sourceforge.net/.

Wordpower is developed using Java and JSF. Like Conceptpower, it uses Wordnet 3.0. However, in contrast to Conceptpower, Wordpower groups all meanings of a term into one entry and provides a URI for that "group entry." For example, there is one entry for the term "horse," which has several definitions, such as "solid-hoofed herbivorous quadruped domesticated since prehistoric times" or "a padded gymnastic apparatus on legs," and is identified by the URI http://www.digitalhps.org/dictionary/c53ef0a2-8950-4035-a828-cc619ba31d34. If a word or term is not included in Wordpower, it can be added through Wordpower's website.

A term in Wordpower has the following properties listed in Table 9.

Table 9: Properties of Terms in Wordpower

PROPERTY	DESCRIPTION
Term	The "name" of a concept (for example "cat" or "Donald Duck").
URI	The URI that identified a concept.
Id	An internal identifier for a concept that uniquely identifies an entry in Conceptpower.
Word	The term that is represented by a Wordpower entry. This should be the normalized spelling of that term.
Id/URI	An identifier that uniquely identifies an entry in Word-power. The id is the last segment of the URI of an entry. For example, if the URI is http://www.digitalhps.org/dictionary/c53ef0a2-8950-4035-a828-cc619ba31d34, the id is c53ef0a2-8950-4035-a828-cc619ba31d34.
Wordnet Id	If a term is contained in Wordnet, this is its id in Wordnet.

PROPERTY	DESCRIPTION
POS	Part of speech of a term (noun, verb, adjective, adverb, or other).
Description	Description of a term. The description is intended only for use by human users.
Vocabulary	List of terms that a term belongs to (each term can only be in one list).

Terms and Vocabularies

Each term in Wordpower belongs to exactly one list of terms, so-called *Vocabularies*. As in Conceptpower, however, these vocabularies are for organizational purposes only. They can be used to group terms by, for example, project or type. Every term in WordNet belongs to a list called "WordNet." In contrast to Conceptpower, there are no WordNet Wrappers. The reason for that is that there are no additional properties, such as a type, attached to entries in Wordpower. This is because the main purpose of Wordpower is to provide a guideline for how terms are spelled rather than what their definition is. Accordingly, there can only be one entry for a word representing different kinds of meanings. Wordpower does not allow adding a word twice.

Web API

Wordpower provides a web API for other applications to request data. An application like Vogon can search Wordpower's database or request information about a specific entry through that API. For example, a request to a URL such as http://my.server.edu/wordpower/rest/WordLookup/horse/noun will return noun words that contain "horse." The URL

http://my.server.edu/wordpower/rest/wordpower/rest/Word/XID-horse-n will return information about the Wordpower entry for "horse" identified by the id XID-horse-n.

4.7.4 Quadriga

Quadriga is a web application that was designed as a repository for storing, managing, and publishing quadruples. It also provides project management functionality to support quadruple annotation projects. Quadriga can be accessed by human users through a website, or by other applications through a web API. This section describes the general structure, features, and implementation of Quadriga. The current version of Quadriga and its code can be obtained from http://quadriga.sourceforge.net/.

Quadriga's prototype was developed using JSF but was later ported to the Spring Framework. It uses a web application called QStore4S[62] to store annotation networks consisting of Appellation Events and Relation Events, and a MySQL database[63] to store other persistent information. QStore4S is an application developed by the Digital Innovation Group that uses the graph store Neo4j[64] for storing network data. The texts that users annotate are stored in a DSpace repository, to which Quadriga has access.

Quadriga consists of five major components: project management, concept list management, term list management, network editing, and project sites. For all of these components, except project sites, an account in Quadriga is required. Quadriga uses an LDAP server to authenticate its users, but it stores user specific details in its local database. This has the advantage that if Quadriga is included in a system with several components that require a login, the same account (username and password) can be used for all components.

---

[62] Further information can be found at http://store4s.sourceforge.net/.
[63] See http://www.mysql.com/ for more information.
[64] See http://www.neo4j.org/ for more information.

Moreover, if there is already an existing LDAP server with registered users, those users can directly login to Quadriga with their existing username and password combination. The next sections will describe each component in detail.

Project Management

The project management component provides functionalities to manage annotation projects in the Quadriga-System. The biggest units in Quadriga are projects, which represents the annotation projects of a researcher or group of researchers. Several people (Quadriga users) can collaborate on a project, which means they can annotate texts and add the resulting annotations to it.

A project contains one or more workspaces. A workspace holds a list of texts that have already been annotated or that are still to be annotated, and all the annotation networks already submitted for the texts in the workspace. Workspaces can be used to organize texts, for example, according to annotators (one workspace per annotator), or according to text topics.

Concept List Management

The main purpose of the concept list management component is to simplify the use of Conceptpower in an annotation project. Vogon requires users to first add the concepts needed in the annotation process of a text to a concept list in Vogon, before those concepts can be used in an Appellation Event. This means that when a user switches to a new project or a new set of texts, all necessary concepts have to be added again. The number of necessary concepts grows quickly with length and number of texts, which can hamper the ease of the annotation process quite dramatically.

The concept list management component is designed to help alleviate this problem. It allows a user to create lists of concepts imported from Conceptpower that then can be used in the different projects of the user. The concept lists created this way can also be shared with other users. For example, user A could create a concept list containing concepts related to Albert Einstein, and a concept list containing concepts related to World War II. In a project concerned with the role of Albert Einstein in the Manhattan Project, user A could import both concept lists into Vogon, which would provide many concepts needed for creating Appellation Events. In contrast, in a project about Albert Einstein's life, only one of the concept lists might be useful. User A could also share his list about World War II with user B, who might be interested in the role of Russia in World War II.

Term List Management

The term list management component serves a very similar purpose as the concept management component. Instead of creating lists of concepts that can be used across different projects, a user can create lists of terms imported from Wordpower. Like concept lists, those term lists can be shared with other users.

Network Editing

The network editing component provides a workflow for reviewing and commenting on networks. This workflow was designed to prevent incorrect networks to be added to the repository. "Incorrect" here means mistakes such as an obviously wrong concept referred to in an Appellation Event (for example, the concept "Hillary Clinton" was specified as interpretation for the term "Albert Einstein"), or structurally wrong relationships (for example, the predicate is used as subject). Especially in the beginning, when users are still in

the training phase, such mistakes are common. The editing workflow consists of the following steps:

1. A user submits a completed network (finished the annotation of a text) to Quadriga for review. From this point on, the user cannot change the network, but has to wait for an editor to review it.
2. The submitted network gets assigned to an editor of the project.
3. The editor reviews the network and creates annotations if improvements or changes are necessary. Those annotations can be attached to single Appellation Events, Relation Events, or whole statements.
4. The editor approves or rejects the network.
    - If no changes are necessary, the editor approves the network. The network gets added to the Quadriga main repository, which contains all approved networks.
    - If changes are necessary, the editor rejects the network. The user can download the editor's annotations for the network to Vogon and modify the network accordingly. When the modifications are made, the network can be resubmitted and the editing workflow restarts.

Project Sites

Each project in Quadriga has a public project site. This project site serves as publishing platform for the networks created in the context of a project. Each network that has been approved by an editor is considered being published. It is assigned a stable URL, so that it can be linked from other websites or publications.

The creator of a project in Quadriga can choose how visible a project is, which dictates who can access the networks in a project. A public project can be accessed by anyone. A private project can only be accessed by the project members. A user might choose to make a project private rather than public, for example, if the annotated texts in the project are highly copyright restricted, so that even networks created for those texts cannot be made public.

Project sites could also serve as a publishing tool for analyses of networks. Future development of Quadriga could include network analysis plugins that users can run on a set of networks. The results of the analyses could be made accessible through project sites. For example, there could be a component that allows users to compare two networks. The results of running this component on selected networks, including information about the networks that were compared, could be published via a project site.

4.8 Conclusion

In section 2.2, I examined the current project landscape in digital History and Philosophy of Science (HPS). I categorized projects and evaluated the promotion and possible reuse of tools that can be used for computational analysis of sources. I identified several issues that, to my understanding, hamper the successful adaption of digital HPS tools and stand in the way of progressing in certain areas of the field. I believe that the Quadriga System avoids several of the identified problems, as I will describe in this section.

I argued that the current project landscape in digital History and Philosophy of Science seems to be biased towards the presentation and dissemination of sources. Many digital HPS projects focus on digitizing materials, and in cases of textual sources, transcribing them. Some projects concentrate on creating contents by writing articles, or

texts for webpages. In all of these projects, however, the main purpose is to publish materials online in forms such as digital collections, or virtual exhibitions. Only a small number of projects use computational methods, which are mostly (semi-)automatic text mining techniques, to aid in the research process.

In addition, the development of research tools is often highly specific to a particular project. Such tools can therefore not easily be applied to other projects. If at all, a computer scientist is required to modify them. For example, the software "Anteater" developed for the project "Science under Scrutiny" can be used to extract specific information (such as who requested a permit to study what endangered species) from Federal Register documents. However, if a project has slightly different requirements, like a different set of texts that need to be analyzed, the software would need to be adjusted. Similarly, although the InPhO project makes all of its code available online, programming skills are needed to apply it to other projects.

It can be positively noted, however, that several projects have open-source policies in place and their source code is freely available. This allows other projects to potentially use existing work and adapt it. What is often missing is documentation on how to use tools and their code. The architecture of a piece of software, its different components, or how it can be extended is in many cases not clear. This situation hampers the successful adaption of existing tools.

The Quadriga System provides computational tools to be used in research projects that are not tied to a specific project, but can be applied to any project aiming to create networks of concepts. Using Vogon, any kind of network can be created; for example, a network between people and places can be generated the same way as a network of scientific

theories and their creators. The graph-mapping feature of Vogon, especially, makes the software highly applicable to the creation of any kind of network. A text can be annotated with a network in Vogon and then transformed into any format to be visualized in a network program such as Cytoscape or on a website.

Networks created using the Quadriga System can also be used to explore text corpora. By linking Appellation Events and Relation Events to specific positions in text, the system captures the necessary information to connect nodes and relations in a network to particular texts. This feature allows for the creation of text exploration tools that go beyond keyword search or tagging of texts. Rather than searching for terms that might indicate a certain relationship between concepts, a user can explore a text corpus by browsing through a network of concepts and their relations to each other and select only those texts that mention a specific relationship.

Each piece of software that is part of the Quadriga System is developed and promoted as its own project. Research projects that use the Quadriga System link to the tools, but are clearly distinct from the software development projects. The needs of research projects using the Quadriga System inform the development of the system's software: however, they do not control it. One reason for this is that the software developed for the Quadriga System, although developed in a bundle, can also be used separately. Conceptpower and Wordpower both run independently from the rest of the system. A project that requires an extendable authority file or dictionary that is available online, could install Conceptpower and/or Wordpower. Similarly, Vogon can be used without Quadriga[65]. Smaller projects, that do not want to or cannot contribute their data to a Quadriga repository

[65] The only software Vogon absolutely requires is Conceptpower.

(for copyright or other reasons) can use Vogon without uploading their annotations to Quadriga.

The reuse of components of the Quadriga System relies on the software being freely available online. Each piece of software developed for the Quadriga System is therefore distributed using open-source licenses and hosted on Sourceforge[66], an open-source project repository. Interested users can download the code as well as executable or deployable applications from the projects' Sourceforge sites. Each software component also has its own website to showcase its features and to publish documentation and tutorials. For example, Vogon's documentation page provides a written tutorial and video tutorials for Vogon users, as well as a general overview of the architecture of Vogon for developers. For some projects, such as Quadriga, the documentation is a work in progress. However, every development project has a discussion forum and links to contact information that allows users, as well as developers, to ask questions and provide feedback and comments.

The Quadriga System is being developed in close collaboration with its users. Developers and users have discussed the software tools and their interfaces on a regular basis. This collaboration helped to ease the annotation process as much as possible by incorporating user feedback and suggestions regarding the graphical user interface of the tools. It also supported the development in regards to the detection of bugs in the software. Functions that did not properly work or created unexpected results were found by users and reported to the developers.

The software of the Quadriga System has been developed with a focus on its reuse and adaption. As many parts as possible should be applicable to different projects and goals

[66] http://sourceforge.net/

contributing to the construction of a "toolbox" for the digital HPS scholar. To achieve that goal, however, tools need to be freely accessible, easy to use also for the "non-computer savvy" scholar, and well documented for end users as well as developers. The Quadriga System tries to address all of these requirements, and continues to improve their implementation for a successful adaption by new projects.

CHAPTER 5

APPLICATION

The Quadriga System can be applied to a broad range of projects. This chapter will describe two projects, the EP Annotation Project and the Genecology Project. I will detail what role the Quadriga System played and how the projects used the system. The EP Annotation Project uses the software of the Quadriga System in an exploratory or prototypical manner, while the Genecology Project uses the software applications as an integral part of its workflow.

Hayles suggests that there are two extremes when it comes to "reading." There is close reading that emphasizes the scholarly interpretation and careful examination of sources. This technique, although precise, is also slow, and one scholar can only analyze a limited number of documents (in an interview with Hayles, Gregory Crane estimates the maximum number of books a person can read in his life to be 25,000) [Hayles 2012]. The other extreme is distant reading, which uses computational methods to analyze large sets of documents, often trying to avoid any a priori interpretation[67] [Hayles 2012]. In the middle of these two extremes, Hayles sees projects that explicitly use a scholar's knowledge and interpretation as input for computational method that might result in new or unexpected results that can be interpreted by scholars or might prompt new questions.

Both projects, the EP Annotation Project and the Genecology Project, use the Quadriga System as such a meso-scale method. By closely reading documents, networks are created that, in a second step, allow for the exploration of large text corpora and which

[67] How far this is even possible as the selection of documents alone is often an interpretive act is a discussion that I will not go into detail here.

might raise new research questions. Network analysis methods can be applied to the created networks and might generate new insights into, for example, relationship patterns between scholars in a field or the centrality of certain scientific concepts. In contrast to distant reading techniques that address such questions, the quality of the data is ensured, as it is hand-coded. This, however, requires a "distant reader" of the created networks to keep in mind the interpretation step involved in creating Appellation Events and Relation Events. Every network created in the Quadriga System is an interpretation of a text by a researcher. The specific perspective on a topic and the background knowledge of an annotator influences what annotations they create for a text and therefore how a network is structured. As a consequence, it is crucial for the interpretation of such networks that the original text for which a network was created is accessible, so that relationships between concepts can be validated and retraced.

The Quadriga System implements this requirement by keeping references to the positions in a text for each Appellation Event. This feature facilitates the development of applications that enable users to explore text corpora through networks while at the same time providing a "zoom in" functionality that allows users to examine the documents in which a specific relationship was annotated. Along with a visualization component for networks, such an application is currently being developed as part of the Genecology Project. The application relies heavily on the graph-mapping feature of Vogon. This feature, which started out as being of minor importance to the software, became a frequently used functionality. One reason for that is that the Quadriga System is a dynamic system that is under constant development and the graph analysis and visualization features of Quadriga are not that far developed yet. By exporting networks from Vogon they can be visualized

and examined using network applications such as Cytoscape. Also, projects that are interested in social networks especially can use these applications to run social analysis functions on networks that were transformed into "simpler" graphs and exported from Vogon.

5.1 EP Annotation Project

The EP[68] Annotation Project is a project in the Digital Innovation Group that has the goal of annotating Embryo Project articles by creating annotations that reflect the relationships between articles similar to the relationships that were originally developed for the Embryo Project. The Embryo Project is an online encyclopedia of embryology that aims to document embryo research in the broadest possible way [Maienschein and Laubichler 2009]. Articles in the Embryo Projects "are written and marked up in such a way that they help populating the database with additional objects that have interesting and relevant relationships to the object of the entry" [Maienschein and Laubichler 2009, p. 11]. For example, there exists an article about Hans Spemann that mentions Theodor Boveri and Wilhelm Röntgen. The marked up article has a relationship to Boveri as well as to Röntgen. However, while an entry for Boveri exists, there is no article about Röntgen. By creating a relationship between Spemann and Röntgen, additional "interesting" information is stored and available for use. One motivation for creating and storing such relationships was to be able to easily answer questions such as "Who was a student of whom? " or "Who worked at a particular place? With [what] particular organisms? " [Maienschein and Laubichler 2009, p. 9].

[68] EP is an abbreviation for Embryo Project.

While there was always an idea for how the relationships in the Embryo Project could be used, and a few visualizations have been created to explore the Embryo Project dataset, there is no publicly accessible implementation of an exploration tool based on the relationships in the Embryo Project. The Embryo Project therefore seemed to be an ideal case for an exploratory project regarding the Quadriga System, especially given the following factors:

1. Embryo Project articles were already marked up with relevant relationships. These relationships were mostly created by the author of an article that is very knowledgeable about the topic of his article, and can consequently make well-informed decisions about relevant relationships.
2. A catalog of relationships has been developed for the Embryo Project that restricts what kind of relationships can be used when marking up articles.
3. While the relationships are created as relationships between articles rather than concepts, usually each article represents a specific concept (for example a person or institution). The relationships between articles can therefore be translated into relationships between concepts.
4. All the articles in the Embryo Project have a Creative Commons license, which makes them easily available for use.

The EP Annotation Project involved three steps. First, standard graphs to be used in the annotation process had to be created. This would ensure that all the annotations representing a specific relationship would be structured the same way. Also, those standard graphs could be reused by other projects that were interested in similar relationships

between concepts. Second, Embryo Project articles had to be annotated. Third, the created networks had to be transformed, exported, and visualized.

5.1.1 Standard Graphs

Because of the structure of Quadruples in the Quadriga System, the relationships that were developed in the context of the Embryo Project had to be transformed into more complex nested Quadruples. The original relationships of the Embryo Project have predicates such as "is teacher of" or "has author." In the Quadriga System, such relationships would not be represented by one predicate but would be expressed using nested Quadruples (see Figure 27). One reason for that is that if the Quadriga System allowed predicates like the ones used by the Embryo Project, its list of predicates would have to be extended with every new relationship. For example, if a project needed the relationship "is mentor of" instead of "is teacher of," a new predicate would have to be added. Using nested Quadruples, the necessary concepts to express the relationship are already present in the list of concepts ("to be," "to have," and "mentor"). Another advantage of this redefinition of relationships is that it allows the Embryo Project to review all existing relationships and make adjustments if necessary. It might be useful to rely on fewer relationships that are well-defined and consistent (for instance, "has author" or "is author of" should be defined as two directions of the same relationship).

For most of the relationships listed in the Embryo Project relationship catalog, an equivalent (nested) Quadruple has been created. All of those Quadruples have been made available as standard graphs through a repository[69]. As more projects start to use the

[69] http://sourceforge.net/p/gobtan/code-doc/HEAD/tree/trunk/NewSG/

Quadriga System, it is hoped that those standard graphs will be used in the annotation process of texts, and that the repository will be extended to include further relationships. The assumption is that the more standard graphs are available, the easier it becomes to start new projects as they can utilize existing Quadruple structures and do not have to develop those themselves.

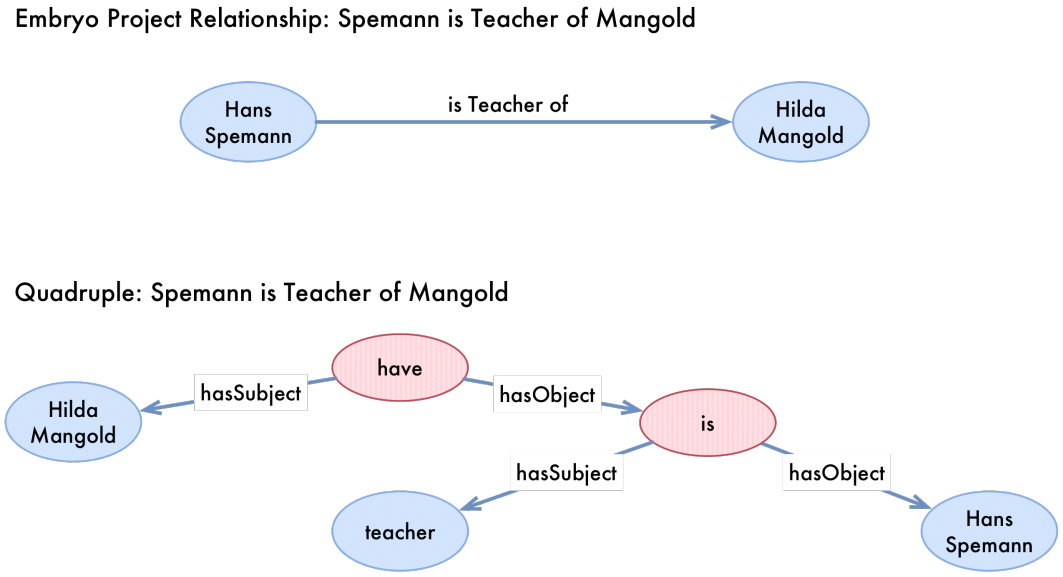

Figure 27: "is Teacher of" Relationship in the Embryo Project and the Quadriga System

5.1.2 Annotation of Articles

As the EP Annotation Project is an exploratory project, it so far has been undertaken as a proof of concept project. There are about 50 annotated articles; all of them describe specific persons (for example Hans Spemann or Viktor Hamburger) rather than institutions or organisms. For each article, about 10 to 20 relationships were created, capturing information such as who was a teacher of whom, who worked with what organism, or what kind of relationship existed between a person and institution. In general, the relationships that were

created were based on the existing relationships already marked up in the texts by their authors. Sometimes, however, additional relationships were generated according to the catalog of Embryo Project relationships.

The annotation process of the EP Annotation Project made it clear that a review system for the Quadruple networks was necessary. Especially in the training phase, users new to the Quadriga System made mistakes, such as switching the subject and object of a relationship or using the wrong concept when there were several meanings available for a concept. It was necessary to review networks to make sure the mistakes were corrected. Such experiences were the basis to the network-editing component in the Quadriga software. This component is designed to alleviate the process of reviewing networks, and to provide a convenient way for reviewers to give feedback.

5.1.3 Export and Visualization

To visualize the created annotations (the Quadruple networks) they first had to be transformed and exported. For this process, mapping files had to be developed that specify how a specific (nested) Quadruple would be transformed. The simplest transformation of a Quadruple is a one-to-one mapping, in which the structure of a Quadruple is preserved in the transformed network. However, such a network is not necessarily ideal to easily answer questions like who taught whom. The Quadruples created with Vogon, therefore, were transformed into simple triple structures in which every node represents a concept of interest (for example a person or institution), and edges represent relationships between those concepts (for instance taught or employed).

Figure 28 shows the network of people and institutions that results from the annotation of about 35 Embryo Project articles. For that network the following relationships were exported:

- A person has another person as teacher.
- An institution has a person as director.
- An institution has a person as student.
- An institution employed a person.
- A person received a degree from an institution.

Figure 29 shows a fine-scale view of the boxed part in Figure 28. Besides providing a fast overview of all the institutions with which James Ebert, William Brooks, and George Corner had some kind of relationship, the network also shows that all three of them worked or studied at Johns Hopkins University. This kind of information can also be retrieved by reading the articles about Ebert, Brooks, and Corner, however, a network like the one shown is a much faster approach. In addition, the information that all three researchers had a connection with John Hopkins can easily be missed when reading the articles as none of them states this fact directly. Ebert, Brooks, and Corner didn't interact with each other directly. However, there are cases in which, although there is no direct interaction, researchers might have indirectly influenced each other by working for or studying at the same institutions. A network perspective like the one shown in Figure 29 might help discover such possibly unexpected relationships.

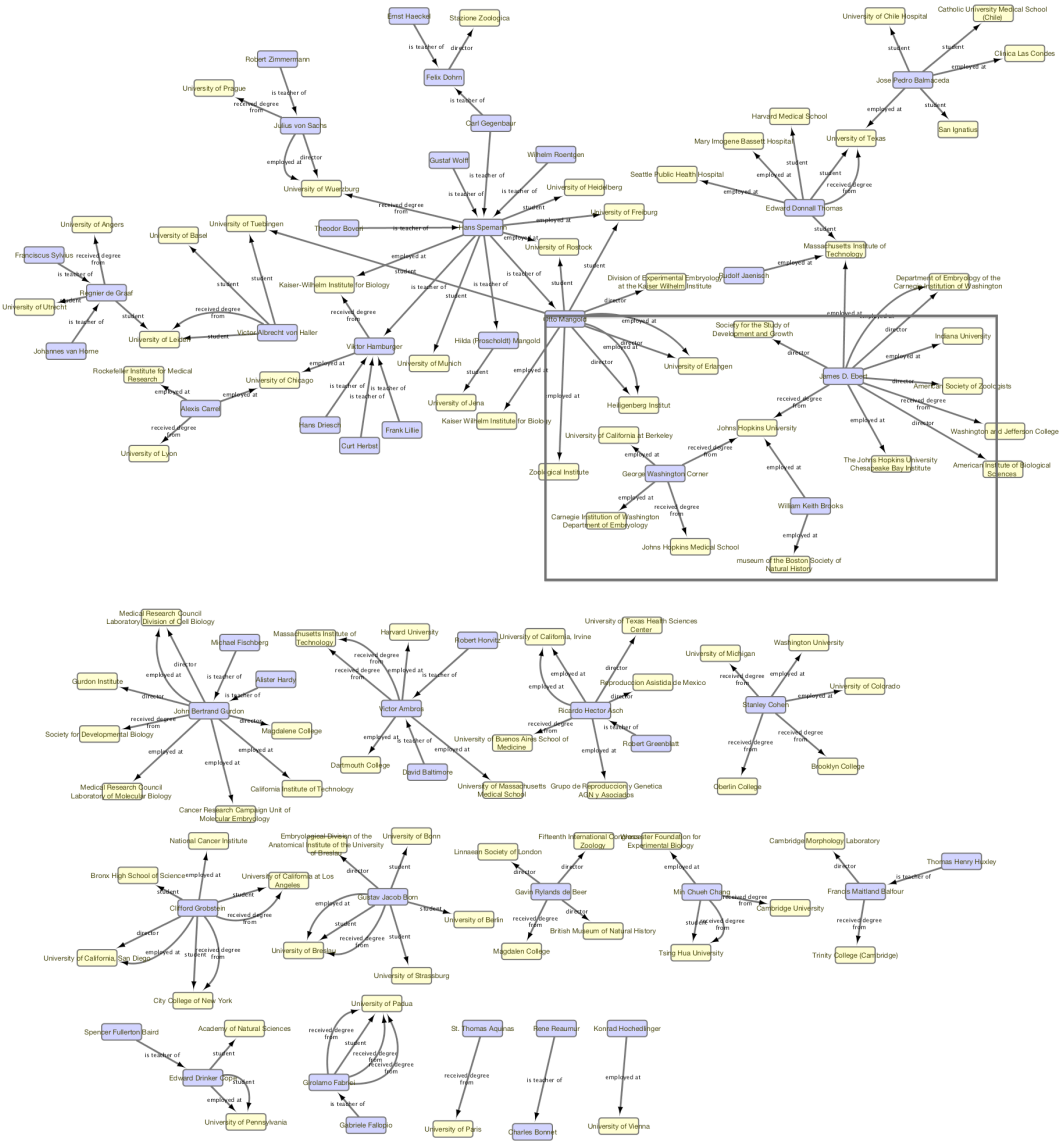

Figure 28: Network of People and Institutions created from 35 Embryo Project Articles.

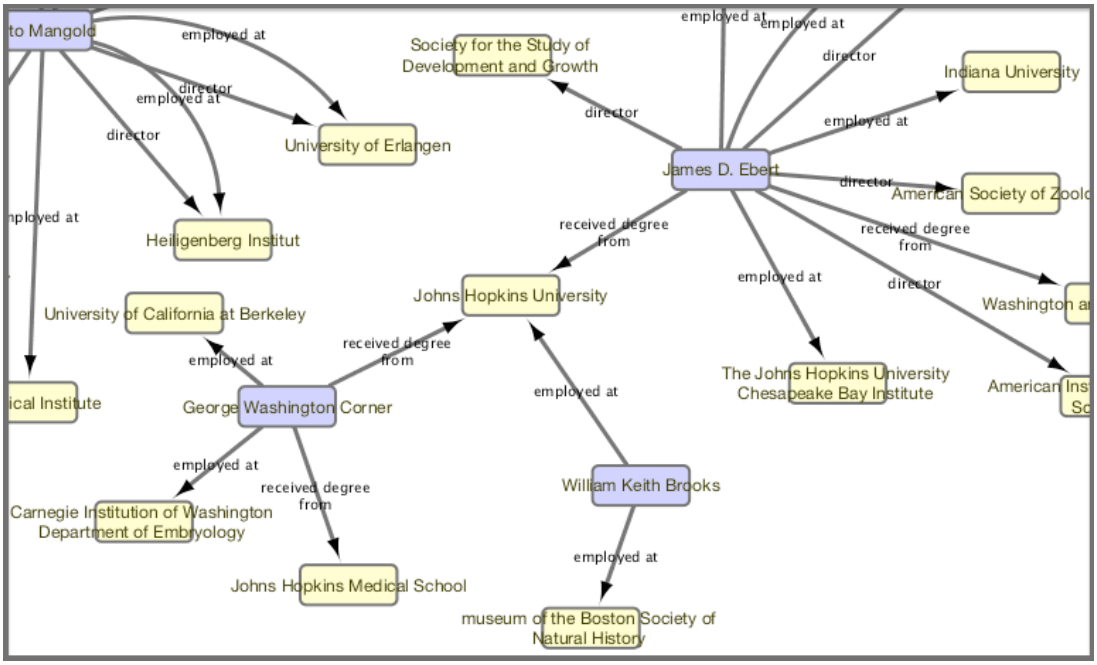

Figure 29: Fine-Scale View of Figure 28

The given example highlights a characteristic and its implications of certain types of networks: networks that do not incorporate time information. The network shown in Figure 28 does not provide any information about when a certain relationship was active. For example, it shows that William Brooks was employed by Johns Hopkins University, but it does not tell when. This could lead to possibly wrong conclusions, for instance that Ebert and Brooks knew each other personally, which is impossible because Ebert was born in 1921, 13 years after Brooks died [Smith 2010, Seward 2008]. For this reason, the Embryo Project Annotation Project started to incorporate time information whenever possible into the used Quadruple structures. However, so far not enough relationships have time information added to them, so that the resulting networks are rather sparse and most institutions, for example, are only linked to one person. If enough time information is provided, it will be possible to generate dynamic networks out of the Quadruple networks.

These dynamic networks visualize how networks change over time. To answer questions such as who was in the same place at the same time, dynamic networks are useful tools.

5.1.4 Conclusion

In conclusion, the EP Annotation Project has shown that various kinds of visualizations of the relationships between concepts in the Embryo Project can easily be created when using the Quadriga System to annotate Embryo Project articles. However, the annotation process is time-consuming. Given that there is no automatic way to transform existing marked up relationships into Quadruples, a human annotator has to create Quadruple networks for each Embryo Project article by hand. Once annotated though, the Embryo Project could greatly benefit from the networks that can be produced based on the annotations. A network visualization of people and places, for example, could be used to explore the research of a specific person or institution and how the focus of research changed over time.

Compared to the existing workflow of marking up relationships between articles, using the Quadriga System has two main advantages. First, time information can be included in networks. By nesting Quadruples, several different kinds of information can be added to a relationship between two concepts; time and place are two of them. This feature could be used to create networks that allow a user of the Embryo Project Encyclopedia to explore its content filtered by time or place. It could also be possible to create timelines of specific places or persons based on the Quadruples. Second, the Quadriga System's feature of storing text positions with every annotation makes it possible to refer back exactly to the point in a text in which a certain relationship was mentioned. For instance, an edge between two people nodes could be traced back to the article that describes the relationship represented

by the edge. A user would have direct access to detailed information about a relation or concept.

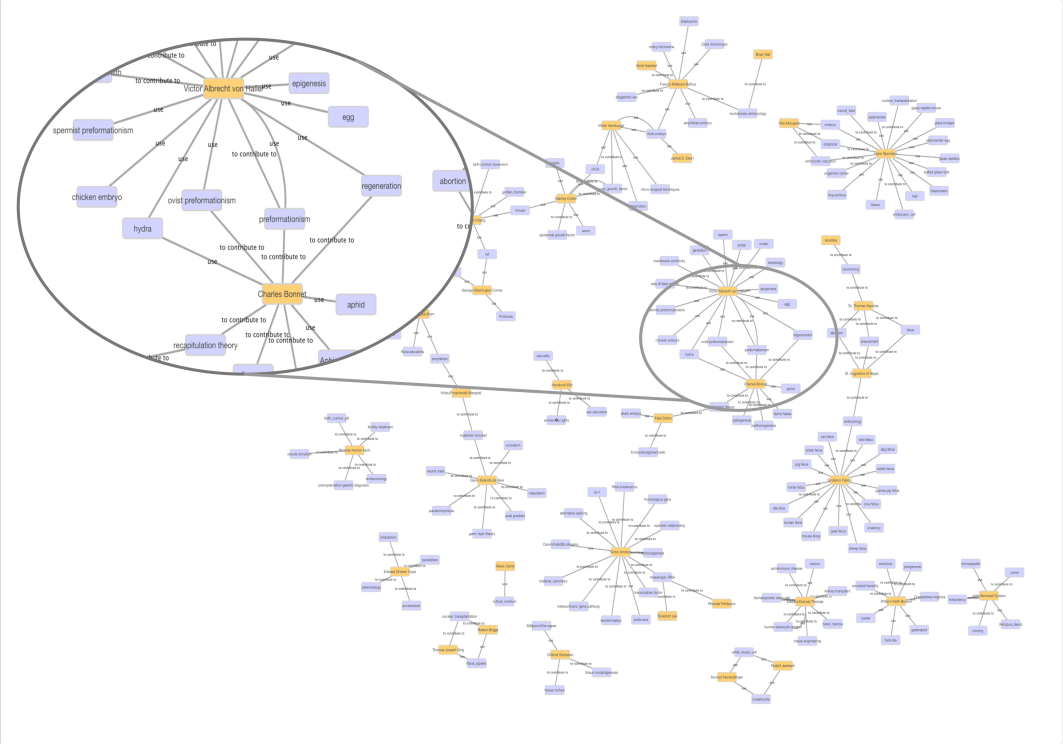

Figure 30: Network of People and Theories, Organisms, and Techniques created from 35 Embryo Project Articles.

A next step for the EP Annotation Project would be the development of a web application that would read the exported networks to visualize them with links to the annotated Embryo Project articles. The application could provide filter and search features for an easy access to the data. Regarding the annotation of texts, so far only articles about persons have been annotated. As the number of annotated articles increases, it would be expected that the connectivity of the network of annotations increases as well, which would enrich the information represented by the network. For instance, Figure 30 shows a network of people and organisms, techniques, or theories on which those people worked. If, for

instance, articles about preformationism were annotated as well and included in this network, there would be more nodes and a higher connectivity between the nodes representing concepts related to preformationism. There would also be an additional node for Nicolas Malebranche, who contributed to the theory of preformationism [Lawrence 2008], but who is not yet represented by an article in the Embryo Project. With an increasing number of articles mentioning Malebranche, the node representing him would be more and more connected to other nodes representing other people, theories, or places. The network resulting from these connections would provide information about Malebranche's life and work even if no article exists yet.

The EP Annotation Project could also be of great use to other projects. The annotations created for the project form a valuable "knowledge base" on which future projects could build. For example, consider a project that focuses on John von Neumann's work regarding computer science. Such a project could use the existing Quadruples of the Embryo Project by using the same authority file as the EP Annotation Project. This way, Quadruples that are created for the von Neumann project connect to the Embryo Project network. A possible von Neumann project could then use this additional data that already exists for its own research. For instance, a network that represents how concepts and theories were influenced by researchers could not only include computer science concepts, but biological concepts as well. Such a network could show how these two fields of study are related, where they connect, and how interdisciplinary exchange could have happened.

5.2 Genecology Project

The Genecology Project studies genecology research in Great Britain during the 20th-century. Genecology is a branch of ecology that studies how genetic differences in plant

populations relate to "geospatial variation in environmental factors (e.g. soils, altitude, climate)" [Peirson et al. 2014, p. 3]. This section describes the first phase of the Genecology Project. One focus of the project lies on the collaboration of researchers in the field of genecology. It asks questions such as who collaborated with whom and where the involved researchers were located. One method to answer these questions is the creation of collaboration networks. These can be visualized and mathematically analyzed to explore collaboration patterns among researchers. Moreover, if geographic information is attached to certain nodes, such networks can be plotted on a map for geographic visualization. [Peirson et al. 2014]

The first phase of the Genecology Project consists of three main steps:

1. Obtaining all the relevant materials for the project in plain text.
2. Annotating the texts in Vogon.
3. Exporting the annotations (Quadruple networks) for visualization and mathematical analysis.

5.2.1 Acquisition of Texts

The Genecology Project uses the Quadriga System to create networks of people and institutions. Before the annotation process with Vogon could start, however, the texts of interest needed to be obtained and a text corpus had to be built up. Depending on the time frame the texts of interest were published, this can be a rather difficult task. While recent publications are usually available as PDF files with embedded text, older publications are often only available in paper form, or exist as scans that do not have text embedded. Vogon needs plain text files for the annotation process and PDFs with embedded texts can easily be converted into plain text files.

In the Genecology Project, the texts of interest are publications and archival materials from the mid-20th century. Students from the Biology and Society program working for the Digital Innovation Group retrieved those as PDF files and, if those materials did not have text embedded in them, the students used OCR software to extract plain text from the documents. In a later step, the text corpus was extended by a series of oral histories that were available in plain text format.

5.2.2 Text Annotation

Once obtained, the texts then had to be annotated in a specific way. The first question the Genecology Project is interested in is what researchers worked on, when, for what institutions, and with whom the researchers collaborated. The Genecology Project uses two standard graphs to represent these two relationships (see Figure 31). One standard graph describes an employment relation between an institution and a person. The other standard graph represents a relationship between two people that engage with each other. Using standard graphs ensures that different annotators use the same Quadruple structure, which is important for subsequent transformation and export of the annotations. To study collaboration patterns among researchers the Genecology Project could also have analyzed coauthorship relations, which could have been extracted automatically to a certain extent. However, coauthorship is not the only way to collaborate. Providing materials and resources, or discussions between two scholars over lunch can be equally interesting. The Genecology Project therefore chose to annotate their texts using Vogon and the two relationships described above.

To decide what institution employed a person at a particular time, the affiliation information of the author given in a paper were used. The acknowledgment section of a

paper was annotated to discover work relationships between researchers, such as support for fieldwork or materials. As in this first phase, the project is more interested in the fact that two people interacted, and less interested in the specific kind of interaction, any kind of relationship that was expressed in the acknowledgement section was encoded as an engages with relation. In a later stage of the project, this might be changed, so that the relationship between two people can be analyzed in more detail.

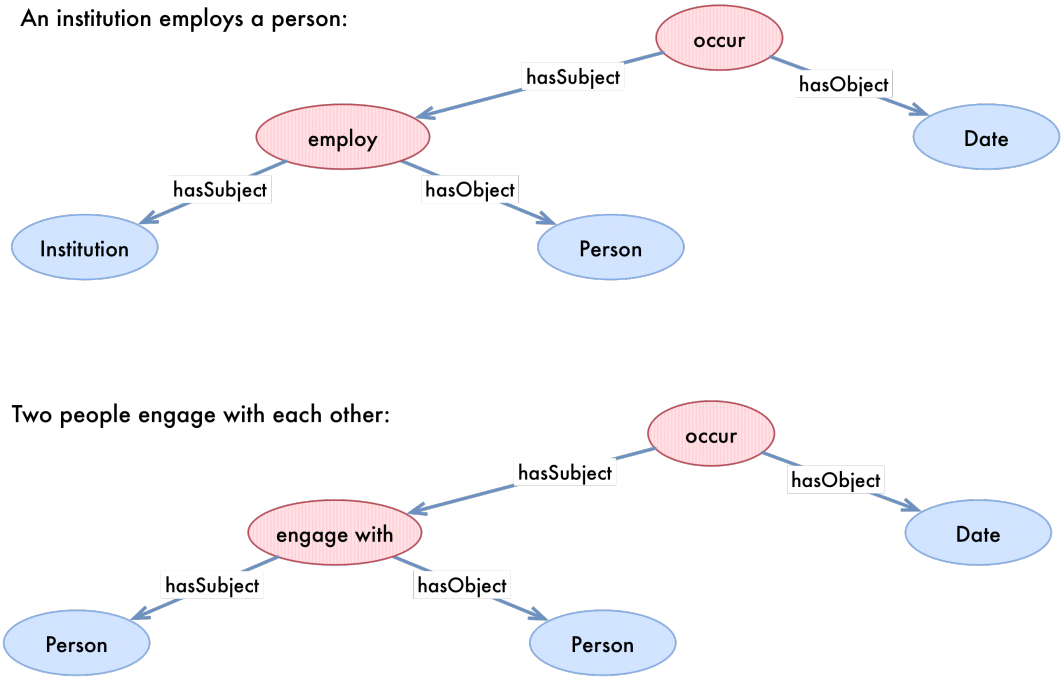

Figure 31: The Two Standard Graphs used in the First Phase of the Genecology Project

The Genecology Project is also interested in when an interaction happened. For example, such information could be used to study whether or not changes in collaboration patterns correspond to conceptual changes in the literature. Does a collaboration that provides a researcher with access to additional resources reflect in the conceptual content of

his publications? Or does a new colleague provide new ideas that impact a person's research?

To be able to answer such questions, time information is attached to the relationships as well. In the project, information of interest is the time a specific relationship is known to have existed, which in the standard graphs is represented by the "occur" relationship. It is assumed that relationships annotated in a publication existed at the time of publication. For example, if a person worked at a given institution according to the affiliation information in a paper, it could be assumed that he also worked there at the time of publication. Similarly, if a person was thanked for his contributions to a project, it is assumed that the author and that person engaged around the time of publication of the paper in which the person is acknowledged. This is an approximation of the actual date two people were engaged or a person was employed, but the underlying assumption of the project is that this approximation is close enough to retrieve meaningful results. For example, if an author thanks a colleague in an article that was published in 1920, then even if the actual interaction happened in 1919, the network resulting from the annotation process and possible interaction patterns detectable in the network won't be falsified by this approximation.

Regarding geographic information, the Genecology Projects makes a compromise between an absolutely accurate representation of information as Quadruples and avoiding introducing so many details into the project workflow that its progress is hampered. Specifically, that means that the fact that institutions might move and change their location has been ignored. It is assumed that there is one location for an institution. The standard graphs used in the project reflect that assumption by having one node that represents the

institution where a person was employed, rather than using a relationship of the form <institution - located in - location>. The information where that institution is or was located is retrieved through the similar to field of Conceptpower. This field contains for each institution a link pointing to a GeoNames entry representing the place where an institution was located. Using that link, longitude and latitude information for a place can be retrieved.

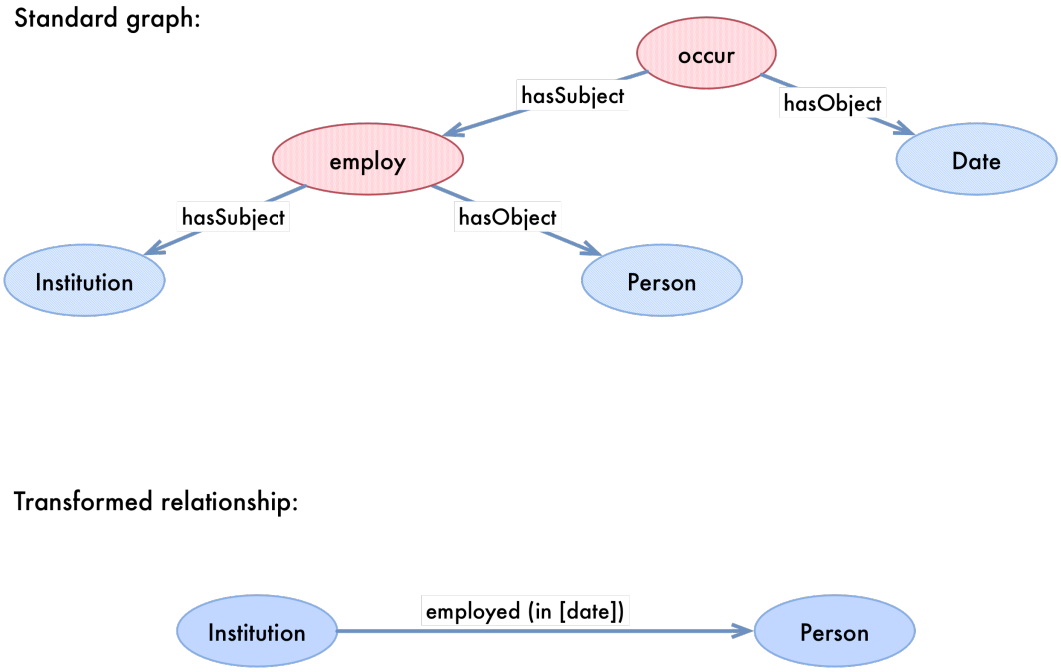

Figure 32: Transformation of the Institution-employs-Person Standard Graph.

The described compromise accelerates the annotation process in the Genecology Project. In general, such a compromise might lead to issues in the long run. As long as institutions did not move during the time of interest, this strategy works. It fails when there is more than one location for an institution. By adding location information to Conceptpower, rather than specifying it by using Quadruples, the information is permanently attached to the concept, and it is not possible to constrain that attachment by,

for example, time. However, in the Genecology Project, for most of the institutions of interest one of the following statements are true. Either an institution did not move, or if it moved it didn't move far, so that the provided GeoNames entry in Conceptpower stays correct. Or an institution moved along with being reorganized, so that can be considered a different institution.

5.2.3 Transformation and Visualization

Besides accelerating the annotation process, however, the strategy of the Genecology Project regarding geographic information also simplifies the visualization of the resulting data. After the texts have been annotated with Vogon, the created Quadruple networks are being transformed and exported similar to the way it is done in the EP Annotation Project. The nested Quadruples are transformed into simple triples that form networks, which can be visualized. One strategy for the transformation of Quadruples is to collapse employment relationships between people and institutions into one node, so that geographic information that is attached to institutions becomes a part of a node representing a person. This strategy is based on the assumption that if an institution at a certain point in time employed a person, then the person can be assumed to be present at the location of the institution during that time. In addition, "engage with" relationships between people are exported as well, so that networks between people that have geographic information attached are generated. Such networks can then be plotted on a map to visualize interactions between people. If time information is exported as well, then the change of interactions between people can also be visualized. A visualization as shown in Figure 33 facilitates quick processing of the displayed data by a viewer and can reveal information that otherwise might stay hidden or might be difficult to detect [Mazza 2009]. For example, the map makes it very easy to see that there

was much collaboration between people in Leeds and London and that many people were based in Exeter and Sheffield that did not collaborate as much. To discover such facts about genecology researchers by "simply" reading texts would be much more difficult.

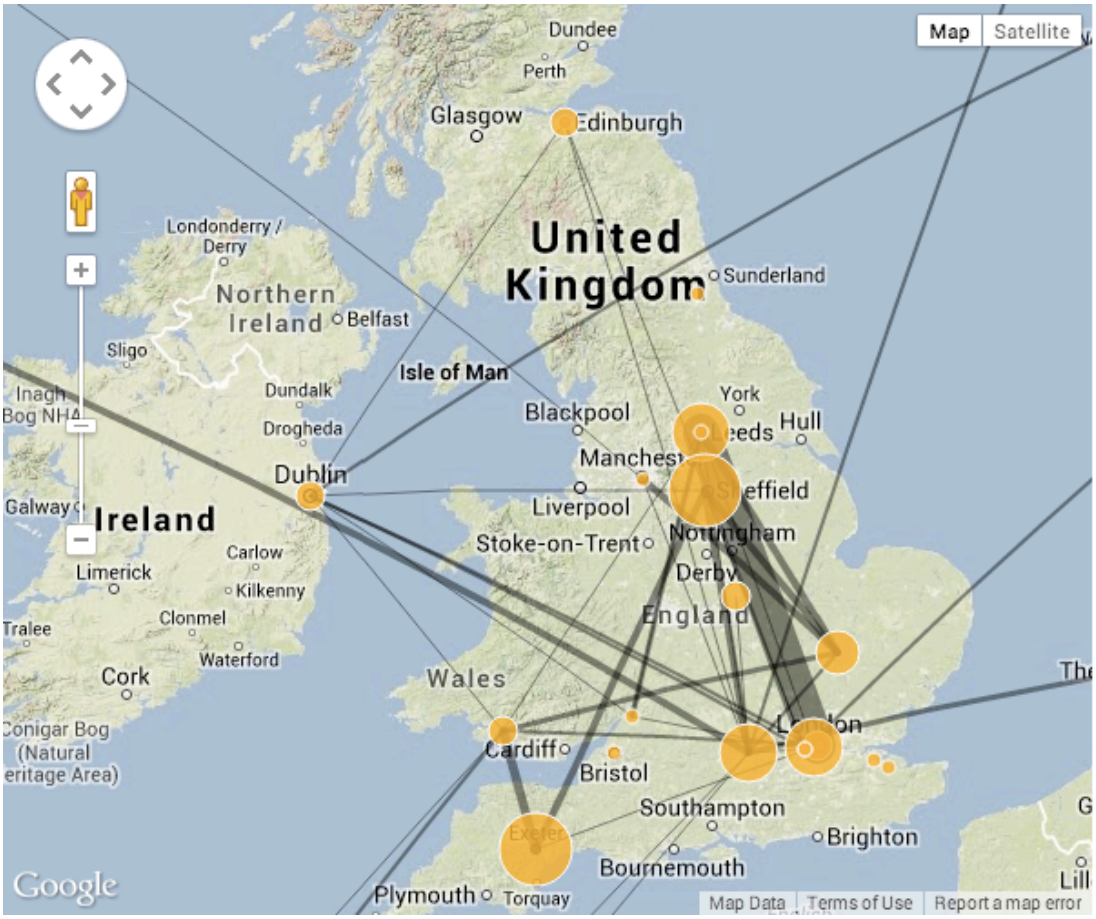

Figure 33: Network of People in the Genecology Project plotted on a Map.

Another strategy to visualize the created Quadruples is to transform the standard graph that represents an employ relationship between an institution and a person into a relationship between institution and person (see Figure 32). The time when a relationship occurred is added as a property to the relationship (or, looking at a graph, an edge). Similarly,

the nested Quadruple representing an "engages with" relationship is transformed by adding the date as a property to the relationship.

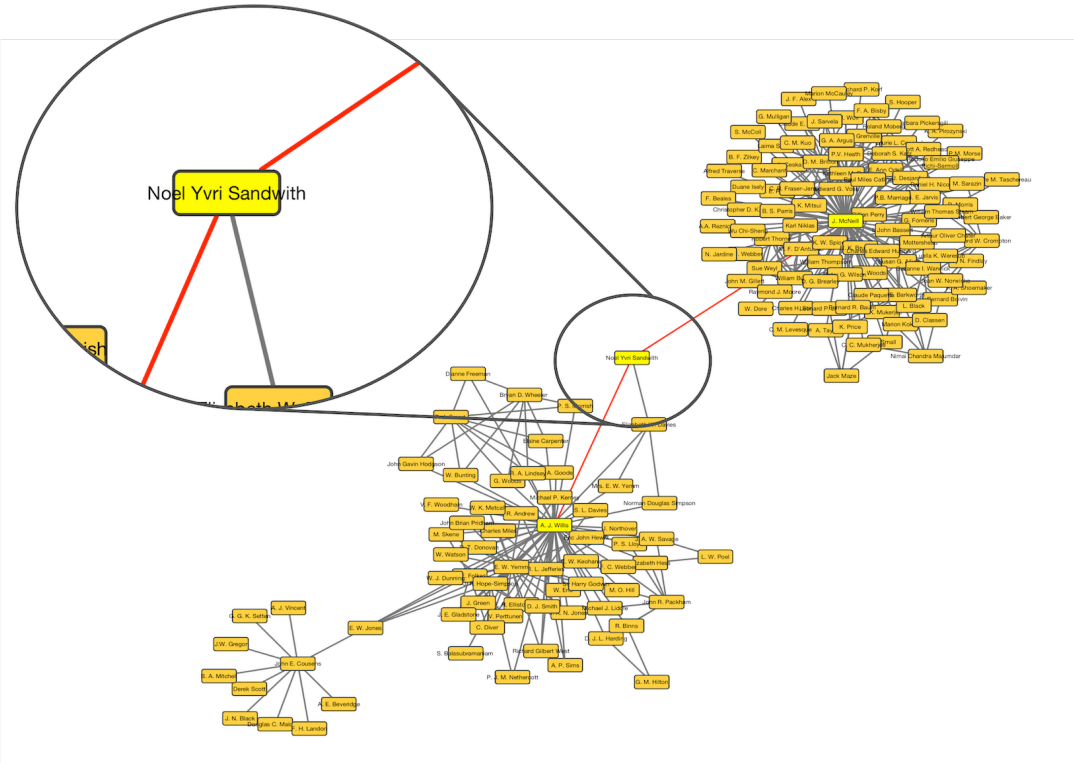

Figure 34: Network of People in the Genecology Project.

The transformed Quadruples form a network such as the one shown in Figure 34. For this network the publications of three researchers (A. J. Willis, J. E. Cousens, and J. McNeill) were annotated and all "engage with" relationships were exported. As seen in the figure, the network can be divided into three subnetworks, with the three authors of the papers being the central nodes of these subnetworks. There are only two nodes that connect the three subnetworks, which means that the authors of the three papers likely did not know each other or did not directly interact professionally with each other. Highlighted in Figure 34 is a node representing N. Sandwith who connects A. J. Willis with J. McNeill. Sandwith

was a botanist in the Herbarium at Kew Gardens in England [Brenan 1966]. Referring back to the annotated texts, Willis and McNeill both visited the Herbarium and worked with Sandwith only a few years apart from each other. Willis mentioned Sandwith in a paper from 1960, while McNeill thanks Sandwith for his work in a paper published in 1963.

5.2.4 Conclusion

It is a question for the Genecology Project whether the connection described above had an impact on the work of Willis or McNeill. Did Sandwith influence the two researchers in a certain way? Or did their work at Kew Gardens provide them with new methods or skills? A detailed analysis of their work might provide answers to these questions. In this particular case, this implicit connection might not be important, however, there will be other relationships that are. Those relationships are rather hard to find by "just" reading publications and other material. Similarly, calculating the influence of a particular person or institution is difficult to do. By creating collaboration networks, such tasks can be accomplished through network analysis methods and visualization of networks. The Quadriga System provides a means to build these networks in a collaborative and structured manner. Vogon allows for the creation of structured datasets that can be used to generate collaboration networks; Conceptpower can be used to eliminate problems such as multiple different spellings of a name that potentially could lead to duplicate nodes for people or institutions; and Quadriga provides a central storage facility for the created annotations[70].

The Genecology Project makes full use of the meso-level features of the Quadriga System. Students closely read individual texts and annotate them with relevant relationships.

---

[70] Although not currently used in the Genecology Project, it is planned to include Quadriga in the project's workflow once its first stable release has been made available.

This ensures the accuracy of the created annotations. Issues such as how annotations are formatted and stored, or differences in the spelling of names are avoided by using Vogon, Conceptpower, and Quadruples. The transformation of Quadruple networks created by all students into "simpler" graphs that form one connected network then allows to ask general questions regarding the created annotations: for example, where were researchers located and what did their collaboration patterns look like? Although these questions can so far not necessarily be answered on the macro-level (due to the rather small number of annotated texts), they could be in the future by adding more annotations from other projects to the Quadriga System. Collaborations of people who are part of the Genecology Project but are not included because they fall out of the scope of the Genecology Project, for example, might still be valuable information for later stages of the project, or for a different research question. The Quadriga System provides an approach for such reuse and combination of data from different projects. Switching back and forth between the "bigger picture," which is represented by combining annotations from hundreds of texts, and the analysis of a single relationship between two entities and how it is expressed in an individual text might be one of the most important features of the Quadriga System. The Genecology Project uses this feature by exporting text position information along with relationships between people and institutions. The links to the original texts can then be used to show users a particular text passage that expresses a relationship when clicking on a specific edge or node in a network realizing scalable reading.

CHAPTER 6

DIGITAL INNOVATION GROUP

The Digital Innovation Group is a group in the Center for Biology and Society at Arizona State University (ASU) with the objective to develop computational solutions for questions in digital humanities with a focus on digital HPS. It trains students from the computer science department together with students from the Biology and Society program in order to develop user-oriented innovative tools, methods, and infrastructures. This group has continued the development of all the software applications implemented as prototypes for this dissertation.

In December 2012, the first students of the Digital Innovation Group were hired and the group was officially named shortly after. The main purpose of the group at that time was to continue the development of the software of the Quadriga System to make the applications stable and easy to use. For that task, programmers were needed to improve the software and users were required to implement a project that would use the software in order to assess its usability. The Digital Innovation Group started out with two Master's students from the computer science department and four Bachelor's students from the Biology and Society program. Under my direction, the computer science students continued to develop the software components of the Quadriga System to improve their usability and functionality. The students from the Biology and Society program started to use the software in two research projects guided by Erick Peirson and me.

It soon became evident, however, that more programming help was necessary to develop the software of the Quadriga System to a point that would make it possible to use it for more than only exploratory projects. By the beginning of the summer of 2013, the group

consisted of fifteen students and then consistently stayed at between fourteen and seventeen students. As of April 2014 the Digital Innovation Group employed fourteen Master's students from the Computer Science department at ASU and three students from the Biology and Society program.

The structure of the group is as follows. The students from the Biology and Society program (the so-called "researchers") are directed by the Head of Research[71]. The Head of Research decides on project assignments and task distribution of the researchers. The computer science students (also called "programmers") are managed by the Head of Development[72] who assigns programmers to projects, decides what technologies are used, and architects software. The Digital Innovation Group is led by the director[73] who provides intellectual guidance and makes decisions about the general development and research focus of the group.

The Head of Research and Head of Development work in close collaboration when setting priorities and to ensure that the software is developed according to the needs of the research projects. This aspect is also the focus of weekly meetings, in which the programmers present the latest changes they made to the software applications to the researchers. These meetings were put in place to discuss usability issues and new or existing features that need improvement or redesign.

Although so far programmers and researchers are student workers, which means that they are paid for their work, the Digital Innovation Group is understood as an educational endeavor. While this was not one of the main goals initially, education became a significant

---

[71] Erick Peirson as of April 2014
[72] Julia Damerow as of April 2014
[73] Manfred Laubichler as of April 2014

part of the group's objectives. Training and teaching of the student workers now play a vital role in the processes of the Digital Innovation Group. Besides hands-on experiences with software tools, programming practice, and teamwork, we have offered professional development workshops for resume improvement and job interview preparation.

This chapter will describe first how I believe the students in the group and their education benefit from their work. This is based on the experiences I had when working with the students, as well as their feedback. Second, I will explain how the field of digital humanities could profit from a concept like the one of the Digital Innovation Group.

6.1 Educational Benefits

The Digital Innovation Group trains students from the computer science department in software engineering and students from the Biology and Society program gain experience in working on digital projects. For both groups of students, the Digital Innovation Group is a setting in which they get hands-on experience with certain applications or frameworks, as well as learn communication and organizational skills such as working as a team, communicating with group members that have a different academic background, or managing a schedule that consists of work as well as school engagements.

Goldberg and White state that to fully prepare engineering students for their professional life, more attention has to be paid to communication and social skills [Goldberg and White 2014]. They argue that many employers criticize that the skills of students are poorly developed regarding working in a team, or collaborating with others [Goldberg and White 2014]. Similarly, Lunenfeld et al. understand the "ability to work collaboratively" [Lunenfeld et al. 2012a, SG14] to be a core competency for digital humanists. In the Digital Innovation Group, the computer science students work in teams on software development

projects and the researchers work collaboratively on research projects. We also encourage frequent cross-discipline communication and discussions in the form of feedback on the software that is being developed, consultation about usability or user interface design, and software or project documentation. We observe that the ability of our student workers to self-organize and work jointly improved vastly since the start of the Digital Innovation Group. While in the beginning the student workers needed a lot of direction regarding distributing tasks in a team and communicating specific project or task details to other team members, they are by now able to work in their teams with much less supervision.

6.1.1 Programmers

Soon after the first computer science students started working for the Digital Innovation Group, it became clear that although they had programming experiences, they were lacking in their software engineering related skills that are needed to develop large software systems (Liu makes a similar observation in [Liu 2005]). For example, they were able to develop code to solve a specific problem: however, in a bigger system the same piece of code would not function properly because they did not consider its context. Also, as the Quadriga System consists of more than five different software components its code base grows rapidly. This means that if the students do not follow best practices, such as adhering to naming conventions of specific programming languages for classes or methods[74], using design patterns for software architecture and code structuring (for example as described in [Starke and Hruschka 2011] or [Freeman et al. 2004]), or observing common documentation

[74] Often naming conventions for a programming language are developed along with a programming language. For example, Oracle describes naming conventions for Java in their Java tutorial (see http://docs.oracle.com/javase/tutorial/java/index.html).

practices[75], the code easily becomes an unmanageable chaos. A lot of these best practices, which have been established in software engineering, had to be and are still being taught.

For that purpose, the weekly meetings in which the programmers present their work from the previous week to the group are followed by weekly meetings in which only the programmers participate (programmers' meetings). In these meetings we discuss best practices, architectural issues, as well as the tasks that need be worked on in the following week. Often these discussions are based on code reviews, in which I audit the programmers' code, as they in many cases do not adhere to best practices. A lot of attention in the programmers' meetings is paid to software design patterns, which are best practices for designing the different components of a piece of software. Being "fluent" in design patterns not only enables the programmers to write more structured, less error-prone code, but also helps them to understand code written by other developers [Tichy 2010], which is an important skill for a software engineer. Another topic often discussed in the programmers meeting are frameworks, or specifically, the Spring Framework and the Eclipse framework. These are used for developing the software components of the Quadriga System. Especially for computer science beginners, the concept of frameworks can be difficult to understand. Demuth et al. position frameworks on the "highest stage on the ladder [of technical knowledge]" [Demuth et al. 2000, p. 1] a software engineer has to climb to become proficient in object-oriented software development. When using a framework, a program's flow is controlled by the framework rather than the developer (a principle called *inversion of control*) [Fayad and Schmidt 1997], which is the opposite of what a programming beginner

[75] Documentation, especially code documentation, can vary depending on programming language, project needs, and documentation tool. For Java, a commonly used tool is Javadoc, for which Oracle provides documentation guidelines (see http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html).

learns when writing his first lines of code. In addition, a developer has to understand the basic structure of a framework and how to extend it with his own code to be able to use it successfully [Fayad and Schmidt 1997].

The weekly meetings are a part of the software development method that is applied in the Digital Innovation Group called *agile software development*. Traditional software development methods are usually based on detailed planning of every step of the development process and a comprehensive description of all the characteristics of the software that is to be implemented [Stoica et al. 2013]. In contrast, agile software development focuses on factors such as easy adaption to changing requirements, constant communication between client and developer team, and the client's satisfaction [Stoica et al. 2013, Beck et al. 2001]. Since its appearance in the late 1990s, an increasing number of companies (among them Microsoft) started following the principles and methods of agile software development [Begel and Nagappan 2007]. By working in the Digital Innovation Group, the computer science students learn the principles of an agile methodology, which includes working on projects with only partially specified or changing requirements, and working closely with the customer (the "researchers").

Besides general programming and software engineering skills, the computer science students in our group get hands-on experiences with widely-used software applications that support the general development process. For instance, the source code of all the projects in the Digital Innovation Group is kept in *version control systems* such as Subversion or Git. These systems manage changes made to the contents of files by keeping track of who made a change to a file and how a file changes [Hinsen et al. 2009]. While most computer science students have learned or read about version control systems, most of our student workers

hadn't used any before they started working for us. After they received an initial training in how to use these systems, they started using them in their work and are now well-trained regarding version control. Another piece of software the programmers of the Digital Innovation Group are using for their daily work is an online software application called "Pivotal Tracker."[76] Pivotal Tracker is used to manage tasks. The student workers have tasks assigned to them. They are responsible for working on those tasks and submitting them for approval when they are done. Often several tasks belong to a bigger component, so that several programmers have to communicate and coordinate their work on tasks. This system teaches the student workers to take responsibility for their work while at the same time improving their collaboration and teamwork skills.

Liu described at the International Conference on Software Engineering in 2005 the design of a software engineering course to teach computer science students a better understanding of software engineering principles and "real-world" projects. He states that "[t]o nurture student's appreciation of software engineering principles and to sharpen their skills of applying software engineering techniques, it is instrumental to expose students to complicated real-world projects in software engineering courses" [Liu 2005, p. 613]. The University of Stuttgart follows a similar approach in their software engineering curriculum. Students work in teams on one-year-long projects, in which they develop software under real-world conditions [Müller et al. 2012]. Müller et al. found that these projects improve their students' social skills as well as their software engineering skills by presenting them with problems and situations that they are likely to experience in their later professional careers (for example customer communication and satisfaction, or team organization). I believe that

[76] See http://pivotaltracker.com/

the Digital Innovation Group has a similar effect on our student workers' education as the courses described by Liu or Müller et al. Our programmers "learn to appreciate software engineering principles and apply software engineering techniques by engaging in large, complex real-world projects" [Liu 2005, p. 614], which makes them more competitive for their professional careers.

6.1.2 Researchers

Lunenfeld et al. list in [Lunenfeld et al. 2012a, Specification 3] a number of "core competencies" they understand as being fundamental for scholars working on digital humanities projects. They sort these competencies into three categories: technical, intellectual, and administrative. Technical skills include knowledge about file formats and data types, familiarity with XML, and experience with interface design. Intellectual competencies are needed to answer questions regarding issues such as the long-term perspective of a project, its extensibility, or its integration into a bigger infrastructure. Administrative skills are required to make decisions about copyright, funding, or long-term sustainability. [Lunenfeld et al. 2012a]

Students from the Biology and Society program, although they are not being explicitly trained in all of these skill sets, gain a certain amount of hands-on experience with some of these core competencies. Their technical skill set improves and enhances as they learn about different file formats, text formats such as XML, digitization processes, or text annotation. For example, all student workers hired as researchers received an initial training in how to find sources using different bibliographic databases and how to use OCR software to extract text from scanned sources. They were taught about the relevance of bibliographic metadata and how to collect it. Regarding intellectual and administrative skills, the

researchers are a central part of the research projects of the Digital Innovation Group (for example, the Genecology Project). As such they learn about the main decisions made for the research projects regarding what kind of materials to use in a project, what role the copyright of those materials play, or about issues related to sustaining projects. Although they are not directly involved in the decision making process, they develop an understanding of the different factors that play a role in the development of digital humanities projects. By collecting materials for the Genecology Project, the students gained practical knowledge regarding questions such as how much time it takes to build a text corpus, or what difficulties need to be taken into consideration.

An experience especially important to students who plan on pursuing a graduate degree and potentially an academic career is the experience to teach and mentor. Since the beginning of the Digital Innovation Group several students graduated and new student workers were hired. This provided a possibility for experienced student workers to teach and mentor new members of the group. The former trained the new student workers regarding the structure of the digitization process and the software involved. One of the researchers was also highly involved in teaching Vogon by running workshops for new potential users. By teaching processes and software, the student workers gained experiences such as the kind of preparations that are necessary to teach a certain topic, how to structure training sessions, and how much content can be taught in one session. A crucial skill gained by the student workers was to communicate complex procedures that involved their own technical vocabulary.

In [Lunenfeld et al. 2012a], Lunenfeld et al. state that documentation is a critical part of any digital humanities project. Although they do not explicitly list user manuals or

tutorials that describe how to use a piece of software, this is an integral part of any software application documentation. The Digital Innovation Group follows the approach that the documentation aimed at the users of a program is best developed by the users themselves. The researchers in the group therefore also write user manuals and tutorials (to a certain extent). The tutorials are not limited to textual descriptions of procedures, but also include video tutorials that provide step-by-step instructions for accomplishing certain tasks, for example, in Vogon. By developing these documentation materials, the student workers not only improve their writing skills, but also gain experiences with screencast[77] recording, video editing, and software documentation tools.

Trilling and Fadel list as core competencies in the 21st century the following three skill sets: information literacy, media literacy, and information and communication technology literacy [Trilling and Fadel 2009, p. 65]. They state that students need to learn how to retrieve and assess information and media, and how to use the technologies that create, provide, and manage that data [Trilling and Fadel 2009]. The Digital Innovation Group supports the process by placing its student workers in an environment in which they work on digital projects along with computer science students. They become familiar with new technologies and they are taught how to use and evaluate those technologies. The students also gain hands-on experience regarding computer science topics, like web application usability, by being an integral part of the software development process.

[77] Screencasts are recordings of the computer screen.

6.2 Benefits for Digital Humanities

> *The problems computer scientists and system implementers have in comprehending the logic of cultural concepts seems to be equally as notorious as the inability of the cultural professionals to communicate these concepts to computer scientists.*
>
> — Doerr [2003, p. 79]

This quote by Doerr captures a problem that has been encountered by several projects in the digital humanities. Claudia Müller-Birn, professor of computer science at Free University of Berlin, for example, describes in an interview about how the approach of computer scientists (or engineers in general) differs from humanities scholars. While the former address a problem often in a highly structured and planned manner, humanities researchers might choose a more exploratory or experimental approach [Müller-Birn 2014]. In teams that consist of both computer scientists and humanities scholars this can cause conflict.

Another example Müller-Birn mentions is the problem of developing user interfaces for software applications. She states that the "mental models" of software engineers and humanities researchers sometimes differ considerably [Müller-Birn 2014]. For example, a software developer might focus on developing a new feature that he would think makes the software more usable, while a researcher might never use that feature. On the other hand, a simple thing like duplicating a button so that it is available in different places of a program can expedite a workflow considerably and might help the researcher much more.

A problem that might be the most significant one for a field like digital humanities is the issue of computer scientists and humanities scholars speaking two different "languages." McCarty asked in a talk how the description of a problem could be "translated" into another discipline and "[h]ow can you do that unless you are or can become a participant-observer of

both? " [McCarty 2007, p. 4]. As Müller-Birn puts it, although the language might be the same, the words used can be very different [Müller-Birn 2014, answer to question two]. For instance, terms like "API", "web service", or "ontology" are either not understandable by non-computer scientists or have a different meaning in another context.

The Digital Innovation Group tries to contribute to closing this divide between the humanities and computer science by placing students from both disciplines into the same environment to collaborate on projects. The students learn on the one hand to explain problems to other students with a different background in a way that avoids technical jargon. On the other hand, they learn concepts from another discipline, which makes it easier for them to phrase questions in a way that the other "party" understands. For instance, an important skill for researchers in digital humanities is to write good feature requests or *bug reports*[78] that describe a problem or enhancement of a piece of software in a way that a developer understands it. It can be observed that inexperienced "bug reporters" often phrase problems from a very narrow perspective, because they are focused on the specific problem they are trying to describe. For example, Vogon received bug reports like the following one: "Eric told me 'vocabulary entry' wasn't supposed to be there." Even with an attached screenshot this is not a very descriptive problem report. Students in the Digital Innovation Group learn how to effectively report an issue by getting an initial training and direct feedback from the developers that ask questions about bug reports if the reports are not understandable to them. For digital humanities this means that students in the Digital Innovation Group are well-equipped with these basic skills needed to work in the field.

[78] A bug report is a short message to a developer that describes a problem encountered while using the developer's software. There are several systems for creating and tracking these bug reports.

Another benefit that the Digital Innovation Group brings to the field of digital humanities is that students from both disciplines become familiar with digital humanities projects and might continue to work in that field. Students from the Biology and Society program learn about the different possibilities and techniques that digital humanities has to offer, while computer science students discover the variety of computationally interesting and challenging questions asked in the humanities. I think that this is an especially important aspect of the Digital Innovation Group. Meeks states that "*nothing is truly built for humanists*; the closest we can get is something built *by* humanists" [Meeks 2011, paragraph two, original emphasis]. I believe that although this situation might be changing, it does pose a problem to digital humanities. Software built for other purposes, although it can be applied to a digital humanities project, might focus on less significant aspects of a project. This in turn might lead researchers to use the software in a way that is not helpful to answering the research questions of the project. For example, Meeks describes a situation in which he taught the network visualization program "Gephi." He tried to explain that the network that the students[79] visualized was an interpretation and should be understood as such. However, his impression was that the students didn't care about that aspect but were much more interested in how to adjust the visualization parameters of Gephi [Meeks 2011]. There are several factors that influence the problem stated by Meeks. There are institutional aspects to be considered. But I believe that an important part is also getting computer scientists and software engineers interested in digital humanities in order to build software specifically *for*

[79] Meeks taught that course at THATCamp, a meeting that brings together humanities scholars and "technologists."

humanists through close collaboration that provides software developers with constant feedback about functionality and usability.

Another problem that is connected to the one described in the last paragraph is that to successfully develop software for the humanities, not only computer scientists but software engineers are required[80]. Wikipedia[81] describes computer science as follows: "Computer Science […] is the scientific and practical approach to computation and its applications" [Wikipedia Contributors 2014b, paragraph one]. In contrast, software engineering is described as "the study and application of engineering to the design, development, and maintenance of software" [Wikipedia Contributors 2014c, paragraph one]. Generally speaking, while computer science is interested in computational problems and their solutions, software engineering is much more focused on developing software. I believe that the field of digital humanities needs both: computer scientists helping to solve computational problems posed by digital humanities, and software engineers to build software that incorporates those solutions. The Digital Innovation Group trains computer science students in software engineering. It focuses on building stable, easy-to-use software that is well planned and documented so that its development can be continued even if developers change. Software needs to be continuously maintained to fix bugs in the software or simply to keep the software up-to-date regarding operating systems and software platforms. By adhering to software engineering principles, the Digital Innovation Group lays the foundation for this long-term perspective.

---

[80] Software engineers are usually computer scientists, but computer scientists are not always software engineers.

[81] I am citing Wikipedia here as it reflects to my belief in the cases of computer sciences and software engineering a widely accepted description of a subject. I am aware that this is not true for every case.

6.3 Conclusion

In March 2014, Goldberg and White gave a talk at the 45th ACM Technical Symposium on Computer Science Education in which they described an interdisciplinary approach to train computer science students. They state that computer science today has an increasing number of applications in various fields, which increasingly requires students to be knowledgeable in more than one discipline [Goldberg and White 2014]. Rather than trying to train those students in two different fields equally, they propose a course model that focuses on the collaboration between students from two disciplines by developing collaborative projects [Goldberg and White 2014]. This approach of Goldberg and White to bridge the interdisciplinary gap by collaboration is very similar to the concept of the Digital Innovation Group and as described in this chapter, and I believe that digital humanities (including digital HPS) can profit greatly from it.

Another approach to satisfy the need for interdisciplinary trained computer science and humanities students is the one created by Stanford University. Beginning in fall 2014, Stanford offers a new major called "CS+X program." This program combines computer science with a humanistic discipline (for example, English or music) and offers courses that integrate those fields. It is hoped that this program will not only introduce computer science concepts to the humanities but that also computer science might profit from this liaison. [Brown 2014]

Stanford's strategy to the, as I will call it, "interdisciplinary gap problem" is an approach on an institutional level, while the concept of Goldberg and White and the Digital Innovation Group are much smaller in scale and produce or train different skill sets of students. I believe that digital humanities and digital HPS need both kinds of students (and

future scholars). There are situations in which computer scientists or software engineers are required that do not necessarily need to be equally trained in the humanities. For instance, software applications that are already fully outlined can easily be developed by software engineers who have a good understanding of the requirements of digital humanities projects. But there are also projects in which "hybrid" scholars are desired, like the ones the Stanford program will produce, that need to have a solid background in computer science as well as a humanities field. For example, those scholars will have a good understanding of what can be realized from a computer science point of view that might be useful for a digital humanities project.

McCarty states that "[i]n setting out to make things, computer scientists have to ask *dumb questions*, that is, questions which have been kept voiceless because the keepers of the domain of those questions have not thought to ask them" [McCarty 2007, p. 16, original emphasis]. I believe, however, that this applies to students of the humanities as well. They too have to ask questions that might be considers "dumb" from the perspective of a computer science student. The Digital Innovation Group encourages its student workers to not be afraid of asking as in an interdisciplinary environment such as digital humanities (HPS) projects learning about the other discipline is crucial for the success of a project.

CHAPTER 7

FUTURE WORK

The Quadriga System provides a method for meso-scale text analysis. It combines close reading techniques with the creation of large-scale datasets that facilitate distant reading and source exploration. As a result the workflow of projects using the Quadriga System can be rather cumbersome and work extensive. To leverage the full potential of the system, a logical next step would be to embed it in a bigger research system that combines macro-, meso-, and micro-scale methods for the analysis of source materials. Depending on the research question, different methods could be used in combination. For instance, macro-level techniques could be employed to find in a large text corpus the documents that seem to be important for a particular researcher; topic modeling could suggest texts that seem to belong to a certain topic, or citation analysis could point to articles that seem to have influenced subsequent research. Meso- and micro-scale techniques could then be employed to examine the selected sources more closely. For example, articles identified through citation analysis could be annotated with Quadriga to create collaboration networks of researchers or to create conceptual networks of a specific research topic.

The European ELIXIR project[82] could serve as a model for such an infrastructure for digital History and Philosophy of Science. ELIXIR is a project with the goal to develop an infrastructure for biological information [Brunak 2012]. The project is a collaborative endeavor of several European countries such as Denmark, the Netherlands, and the UK with its "hub" in Hinxton, Cambridge, UK [ELIXIR 2014]. It addresses the problem that every year an increasing amount of biological research data is produced by various

[82] See http://www.elixir-europe.org/

laboratories and companies around the world; storing, managing, sharing, and analyzing such data is not a trivial task [Marx 2013]. While there might still be a difference in data volume produced every year by the life sciences and the humanities or digital HPS, there are certain similarities between both fields. Similar to the humanities that deal with data such as texts, images, video or audio materials, data in biological research is very heterogeneous (for example, gene sequences or medical records) and hard to standardize [Brunak 2012]. In both fields, data is created in many different places, stored and used in various ways.

ELIXIR aims to "help scientists across Europe safeguard and share their data, and to support existing resources such as databases and computing facilities in individual countries" [Marx 2013, p. 257]. According to Brunak, however, ELIXIR does not only focus on data sharing but also on developing an infrastructure for tools to analyze shared data. He envisions such an infrastructure to facilitate the discoverability, benchmarking, and interoperability of such software tools [Brunak 2012]. In the life sciences, tools are typically run in succession creating "processing pipelines" that ELIXIR is planned to support [Brunak 2012]. Similarly, a digital HPS infrastructure could provide a platform for creating pipelines of macro-, meso-, and micro-scale tools that aid the analysis of materials. As part of ELIXIR, the European Bioinformatics Institute (EBI) in Hinxton, UK, is developing "Embassy Cloud," a cloud service that allows researchers to analyze their data and "run data-driven experiments" on EBI computers from anywhere [Marx 2013]. This allows researchers, even if they don't have the necessary computational resources themselves, to run experiments on their (or other's) data [Marx 2013]. A similar situation may emerge in the humanities and digital HPS. Certain (especially macro-level) techniques require a lot of computing power for which many scholars are not equipped. A cloud-based infrastructure,

which makes text analysis tools available to researchers around the world, could facilitate the application of computational tools to digital HPS research.

The European DARIAH project (Digital Research Infrastructure for the Arts and Humanities) is an endeavor in the field of digital humanities similar to the ELIXIR project. It aims to support the work of humanities scholars by building an infrastructure that assists in creating and using research data as well as software applications. DARIAH puts a special emphasis on facilitating the application of digital and computational methods and the creation of an infrastructure for the "federation of knowledge," which refers to digital collections hosted by various European institutions such as museums or libraries. The project's approach is to provide an entry point to different kinds of data sources without imposing a general metadata schema or building one overall repository, but instead to incorporate the heterogeneity and loose coupling of resource providers as a fundamental feature of the system. The ultimate goal of DARIAH is to provide services that enable scholars to query and analyze the federated materials in an integrative manner. [Henrich and Gradl 2013]

A similar project to DARIAH is CLARIN (Common Language Resources and Technology Infrastructure), which focuses mainly on language technologies for the humanities and social sciences. CLARIN's goal is "to turn the existing, fragmented technologies and resources for processing and analysing human language into accessible and stable services" [Wynne 2013, p. 90] by creating a distributed system that integrates various sources and services and makes it easy for any scholar to link different tools and data using a regular desktop computer. CLARIN envisions a system in which a researcher logs on to the

system once to then build text corpora and use a number of different tools to analyze them. [Wynne 2013]

All projects described above are concerned with the problem of dispersed resources ranging from data to computing resources. Marx asserts that "[i]f he or she [a researcher] lacks the computational equipment to develop it [an idea], he or she might not even try" [Marx 2013, p. 257]. This statement about big data research in the life sciences can be directly transferred to research using computational methods in digital humanities and digital HPS. Models to alleviate this problem such as DARIAH and CLARIN could be adapted by the digital HPS community, benefitting from their work and experiences. These projects would also be good partners to interface with in order to extend the source and tool selection that could be provided through a combined infrastructure. Due to the relationship of HPS to the sciences, however, collaborations with projects such as ELIXIR might also be desirable. The data that is offered by such projects might be of interest for particular research projects in the history of science.

An infrastructure for digital HPS would consist of several layers and components with different requirements and functionalities. There are institutional questions as well as copyright, funding, and resource issues. The conceptual layer needs to be backed up with an implementation of the system that manages different data sources with different interfaces and metadata standards and offers access to various tools that might require specific input formats and produce diverse kinds of data.

In this section, I will detail a number of possible enhancements for the Quadriga System and some ideas for the implementation of an infrastructure for digital HPS. Some extensions are closely tied to the described software components, others are new

components that could be added to the system and might raise questions on the conceptual layer. The sections in this chapter are sorted in the following way: First, I will describe some concrete extensions that could be made to components of the Quadriga System. Next, I will describe enhancements to the system that are extensions to existing software components that are less straight forward and need further research. Last, I will describe how the Quadriga System could fit into a bigger system as part of a general workflow for text analysis.

7.1 Software Enhancements

Regarding the development of the different components of the Quadriga System, there are a few logical next steps that would integrate the system further into the Semantic Web and facilitate its applicability to future projects. This section details concrete extension ideas for Conceptpower, Quadriga, and Vogon.

7.1.1 Linked Data Extensions

As described in 4.7.2, Conceptpower follows three of the four rules Berners-Lee defines for Linked Data. A first logical step would therefore be for Conceptpower to comply with all four rules. For that it would have to use standards such as RDF for answering queries or to provide the possibility to query Conceptpower using SPARQL.

For instance, currently Conceptpower returns the information about requested concepts in XML format, not in RDF form. Appendix B.1.2 contains an example response. Listing 9 shows how the same response could be formatted in RDF.

Listing 9: Conceptpower RDF

```
1  <rdf:RDF
2        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3        xmlns:dhps="http://www.digitalhps.org/rdf">
```

```
4
5       <rdf:Description rdf:about="http://www.digitalhps.org/concepts/ CON1c268257-
            3f0e-4059-b4bb-a394ae2ce2a8">
6               <dhps:wordnet_id>
7                   WID-10954498-N-??-einstein
8               </dhps:wordnet_id>
9
10              <dhps:lemma>Einstein</dhps:lemma>
11              <dhps:pos>noun</dhps:pos>
12              <dhps:description/>
13              <dhps:conceptList> Persons</dhps:conceptList>
14              <dhps:creator_id>admin</dhps:creator_id>
15              <dhps:equal_to>
16                  http://viaf.org/viaf/75121530
17              </dhps:equal_to>
18              <dhps:similar_to/>
19              <dhps:modified_by/>
20              <dhps:synonym_ids>
21                  WID-10954498-N-02-Albert_Einstein,
22              </dhps:synonym_ids>
23              <rdf:type
24                  rdf:resource="http://www.digitalhps.org/types/
                        TYPE_986a7cc9-c0c1-4720-b344-853f08c136ab" />
                <dhps:deleted>false</dhps:deleted>
25      </rdf:Description>
26
27  </rdf:RDF>
```

The addition of a feature that returns RDF instead of XML is relatively straightforward. Enhancing Conceptpower with a SPARQL endpoint, however, is a more complex task. Internally, Conceptpower does not use a triple store or another storage solution that stores data in RDF format, so that the data cannot be queried directly using SPARQL. The database used in Conceptpower is an *object database* called "DB4o."[83] Hence, a Conceptpower component that answers SPARQL requests would have to parse the SPARQL query and translate it into database requests that Db4o can answer.

Both of the described enhancements are rather uncomplicated additions to Conceptpower. If realized, Conceptpower could be connected to the Linked Data network. Semantic web projects could use Conceptpower as authority file to refer to specific

[83] See http://www.db4o.com/.

concepts. This could be especially useful for projects that deal with scientific concepts for which there exist no authority files, or lesser-known people that do not appear in VIAF or any other authority file.

If similar additions would be made to the other components of the Quadriga System, this could enable the Quadriga System to gather external information about concepts used for annotating texts, or facilitate the development of new tools. For example, the Embryo Project, the MBL History Project[84], and Quadriga could provide their data according to the Linked Data guidelines. The Embryo Project and the MBL History Project would provide information about the people and places mentioned in the articles and depicted in the pictures. Quadriga could offer an interface that would allow retrieval of all the texts that have annotations using a specific concept. If these three applications would use Conceptpower as authority file, a semantic web browser or another piece of software (in the following called the "semantic web agent") could collect information about concepts via their RDF/SPARQL services and display those to a user. For instance, if a user requests information about Viktor Hamburger, the semantic web agent could get a list of courses and students that Hamburger taught from the MBL History Project, the Embryo Project could supply information about Hamburger's life and career, and Quadriga could provide a list of documents that mention Hamburger.

7.1.2 Federating Conceptpower

Conceptpower is based on the assumption that ideally, one Conceptpower installation (also called *instance*) stores all concepts required in all projects. External authority files can be

[84] http://history.archives.mbl.edu/

linked but Conceptpower contains concepts representing those linked authority file entries. However, there might be situations in which two projects work on different topics and require two different Conceptpower installations (each with its own concept storage). For example, the projects might be using different types or type names, or the projects might be worried that sharing a Conceptpower installation might effect the accuracy of their data. However, one of the projects might have a broader topic than the other one, and the concepts it adds to its Conceptpower instance might be useful for the other project as well.

In such cases, a feature that allows the federation of Conceptpower installations might be desirable. In a Conceptpower federation, each Conceptpower instance would have access to the concepts stored in other instances. When querying one instance, that instance would pass on the query to other Conceptpower installations in the federation. The results from such a federation query would return all concepts matching the query in all federated Conceptpower instances.

To federate Conceptpower installations, two main extensions need to be developed. First, a component is required that passes queries on to other Conceptpower instances and collects their results. This includes filtering the results from other installations to remove duplicates. Second, Conceptpower would need an additional module that allows registering the Conceptpower installations that are part of a federation. This module would also have to manage the mapping of different types used in different instances.

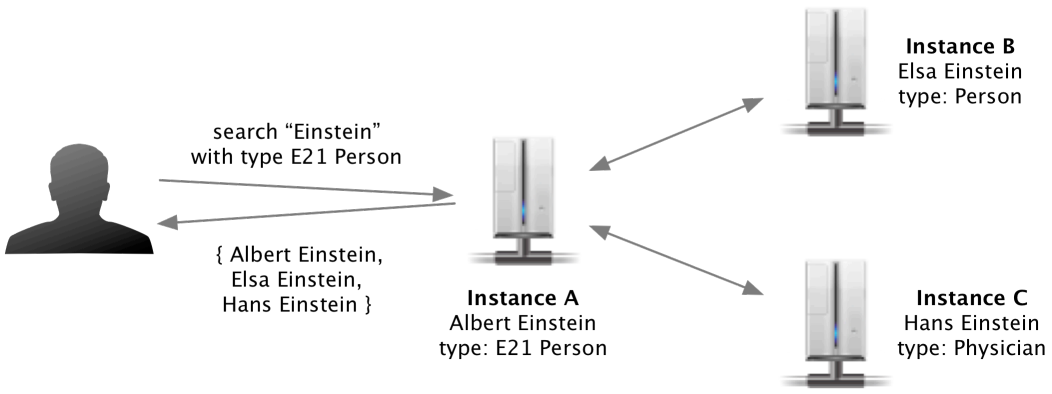

Figure 35: Federating Conceptpower Instances

Figure 35 shows an example of a federation of three Conceptpower instances. When the user searches for all concepts representing a person whose name contains the "Einstein" in instance A, instances B and C are automatically queried as well. The user is presented with the person concepts containing the word "Einstein" from all three instances. In order to query instances B and C, instance A requires a mapping that specifies that the type `Person` in instance B, and the type `Physician` in instance C can be mapped to the type `E21 Person` in instance A.

7.1.3 Graph Mapping

Currently, graph mapping is only available as part of Vogon. A user can map all the graphs consisting of Appellation Events and Relation Events of all the texts in a workspace at once. However, if a project contains more than one workspace, the graph mapping process has to be run on each workspace separately. Also, each workspace has to be downloaded to Vogon first, before the graphs it contains can be transformed. A reasonable next step would therefore be to add a graph-mapping feature to Quadriga.

In Quadriga, such a feature could be available not just for workspaces but also for whole projects. Rather than running a graph transformation on each workspace individually, a user could transform graphs of all workspaces in a project at once. This would have several benefits. First, the transformation process would be less time consuming for the user as the process would have to be started only once for the whole project rather than several times (once for each workspace). Second, the computation that can be quite lengthy is shifted from the client to the server. This means that a user can start a transformation process and get notified once the process has finished. The process is not influenced or limited by the actions of a user, like shutting down his computer or running other computationally heavy operations, or limited by the processing power of a user's computer. In addition, if Quadriga included a graph-mapping component, it could also provide features to manage the transformed graphs. Those features could allow users to store and export the transformed graphs, or to automatically upload them to a different application such as a triple store.

A graph-mapping feature could also facilitate connecting Quadriga to the Linked Data network. As described in Chapter 4, Quadruples are rather complex elements consisting of nested triples, with each part of a triple containing several pieces of information. It might not be practicable to publish Quadruples in their original format to the Linked Data network. However, if they are transformed into "simpler" triples, Quadriga could provide a lot of useful information about the contents of texts. For example, in the EP Annotation Project, published networks could be automatically transformed into graphs using the original Embryo Project relationships. Those graphs could then be made available in RDF format. A process like that would lead to a loss of information; unless reification is used, a relationship cannot contain additional information, like in what text was a

relationship expressed. However, it could still produce useful information that can be published to the Linked Data network.

7.1.4 Text Exploration

So far, Vogon is developed as a text annotation tool. Its functionality focuses on annotating texts. However, Vogon could also be useful for exploring texts and their interpretations. For instance, when a text is added to Vogon, the program could download all networks that were created for that text earlier, or by other researchers. Those networks could be displayed next the text as additional information to the reader (see Figure 36). The user could then use the network to find relationships that he is interested in, and have Vogon highlight the text parts that were annotated with the relationship. Or, the user could use the networks as basis for his own annotations. If there are conflicting relationships the user could find the passages relevant to the conflict to examine it more closely.

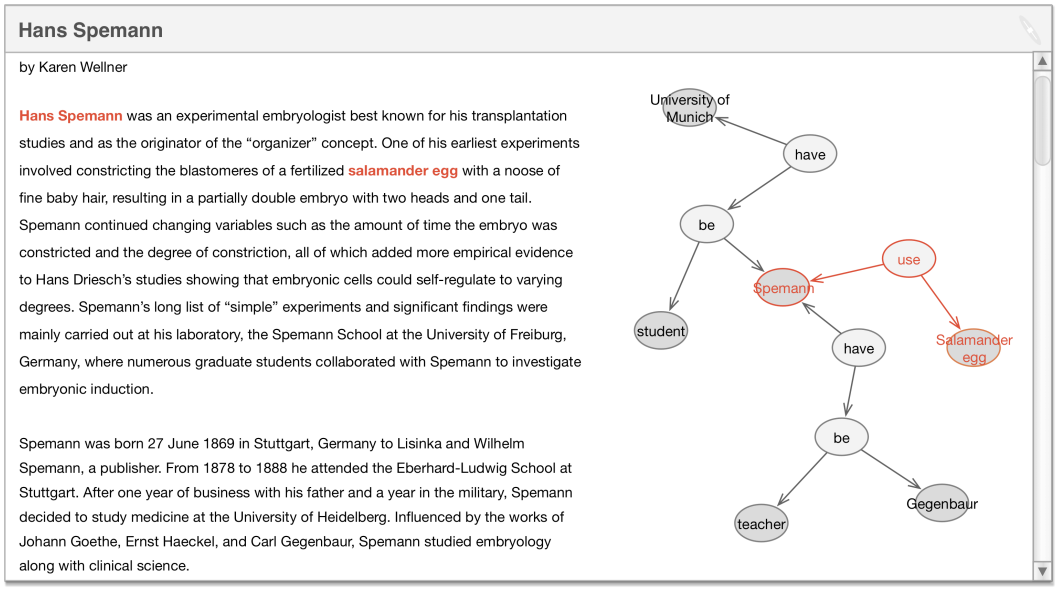

Figure 36: Exploring Texts using Vogon

A network-based text exploration feature could also be developed for Quadriga. Quadriga could offer a search functionality that would allow a network-based approach to documents. In [Cameron et al. 2010], the authors propose a very similar system based on the idea of a "data-centric view of corpora" [Cameron et al. 2010, p. 1]. They suggest that a search based on subject, predicate, object triples in which a user can browse through a network could return much more accurate results compared with the "traditional" full text search in documents, and might be especially useful in cases in which a user has only a vague idea of what he is looking for. A network-based search could facilitate the process of narrowing down the topics of interest much more easily [Cameron et al. 2010].

However, the system proposed by Cameron et al. uses automatic Named Entity Recognition in combination with an existing corpus of triples that describe the relationships between entities but are not rooted in particular texts. In contrast, in Quadriga the text corpus exploration would be based on annotations by researchers and would therefore be more accurate. A user exploring a text corpus would not only be displayed with a list of texts containing a specific concept, but containing a particular statement. A module could be added that implements the functionality described in [Cameron et al. 2010] to be able to explore a bigger, not yet annotated text corpus.

A text exploration module for Quadriga could be extended with a network search component that allows a user to specify a pattern to find in the networks stored in Quadriga. For example, a user might want to find all texts that contain a Quadruple of the form "someone is a teacher of someone else." Quadriga could provide a SPARQL interface to search for all the Quadruples that match that pattern. In addition, Quadriga could offer a graphical search interface in which the user builds a graph with placeholders that represent

that query. Chau et al. describe a graphical user interface for querying networks, which could be very useful in the Quadriga System. Using their application GRAPHITE, a user can construct a network query by specifying nodes, their attributes, and edges. The program tries to find graphs that match the query exactly, and also returns near matches, in which, for instance, two nodes are not direct neighbors but separated by one node [Chau et al. 2008].

7.2 Future Research Topics

There are two concrete research topics that the Quadriga System could benefit greatly from: extracting Appellation Events and Relation Events automatically from texts, and analyzing networks of Appellation Events and Relation Events. In this section, I will detail those two possible research questions.

7.2.1 Automatic Term and Relationship Detection

Annotating texts in Vogon is a rather slow process. Annotators have to annotate each term that should be part of a Relation Event with an Appellation Event. Sometimes a term has to be annotated several times because it appears throughout the text as part of different relationships. There are many cases in which the annotation process could be facilitated by a module that automatically detects relevant terms and their relationships. However, as the Quadriga System is based on the idea that networks created for texts are interpretations of these texts, such a component could only make suggestions that an annotator has to accept or reject based on his understanding of a text.

In the context of this dissertation, I started to work on a component that would automatically extract Appellation Events. This component presents two main challenges. First, relevant terms need to be identified in the text based on features of the text. Second,

terms that have been selected by the component have to be linked to the concept form Conceptpower they represent. Figure 37 shows the Appellation Event extraction workflow. In the following, I will describe each part of the workflow of the prototypical component I developed in respect to these two questions.

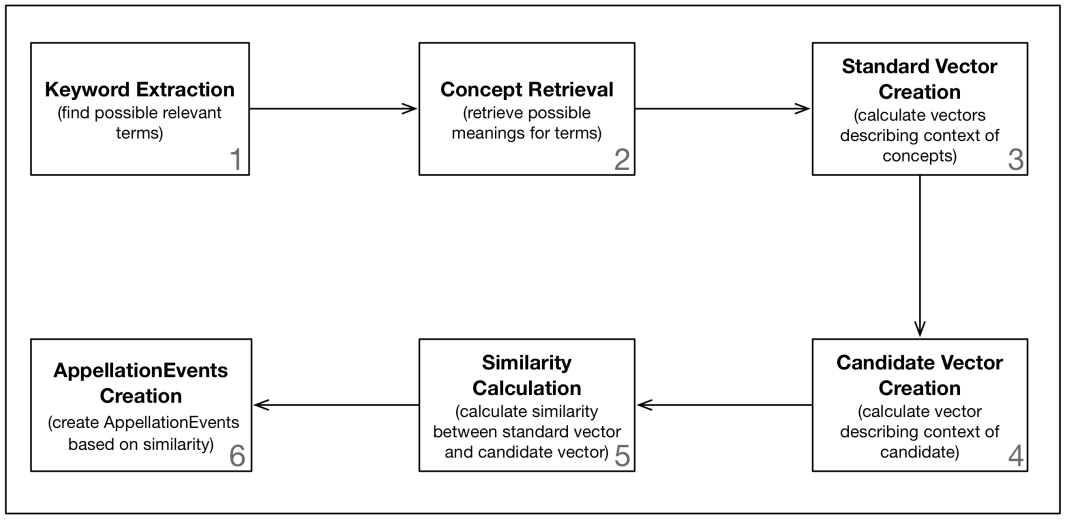

Figure 37: Appellation Event Extraction Workflow

Step One: Keyword *Extraction*

The Appellation Event extraction component I implemented combines four different approaches to identify relevant terms in a text. It uses three different libraries, as well as a method I implemented that uses existing Appellation Events to find terms in new texts. First, a library called "XtraK4Me" is used to identify keywords in a text relevant to the topic of the text. XtraK4Me was developed in the context of digital libraries to extract key phrases from documents [Schutz 2008]. For the embryo article about Samuel Randall Detwiler[85] (the text can be found in Appendix C), the following keywords were extracted:

[85] http://embryo.asu.edu/pages/samuel-randall-detwiler

Detwiler, limb, Ross Harrison, Samuel Randall Detwiler, vertebrate retinas, neural development, work, lab, embryos, college, anatomy

While this list contains several terms that are relevant to the text, there are a number of other terms that are not listed, for example "Yale Medical School" or "Harvard University." In a next step, the text is therefore analyzed using the Stanford Named Entity Recognizer [Stanford NLP Group 2014]. This library finds, in addition, terms such as:

Samuel Randall Detwiler, Detwiler, Viktor Hamburger, Ironbridge, Pennsylvania, Mary Hallman, Ursinus College, Yale University, …

The last library that is used to find terms that might be relevant (also called *candidates*) is GATE, a text processing software [University of Sheffield 2014]. GATE provides a graphical interface for text processing tasks, but can also be called through a Java library. Using GATE, all nouns are extracted from a text and added to the list of possible candidates.

The last step in generating a list of candidate terms for creating Appellation Events is a component that uses existing Appellation Event annotations. This component examines all the Appellation Events created before and produces a list of words or phrases that were annotated. It then tries to find words or phrases that have a high similarity to the terms in that list. To calculate the similarity of two terms, the Jaro-Winkler distance is used. This algorithm calculates the similarity based on the number of equal characters and their order [Naumann and Herschel 2010]. This process ensures on the one hand that terms that were not found by the three external libraries but are frequently annotated are added to the list of candidates. On the other hand, the Appellation Events that annotate terms similar to terms in the text being analyzed can inform the next step of identifying what concepts are referred to in the text.

Step Two: Concept Retrieval

After the component generated a list of candidate terms, it tries to detect what concepts the terms refer to. For each candidate term a list of concepts is retrieved from Conceptpower that contain the term. However, only concepts that are similar to the candidate term (that have a Jaro-Winkler distance score higher than 0.7[86]) are stored as possible concepts. In addition, if there are names for which no concept could be found, the component requests a list of concepts containing only the last word of the name, assuming that the last word is the surname of a person. This rule was created because the expression of a name often varies a great deal. The text might use only the surname to refer to a person while the concept is represented by the full name.

Step Three: Standard Vector Creation

The next step of the Appellation Event detection process is to decide what the "correct" concepts are (which concept from the list of concepts retrieved for a term should be selected by the program), and if a term/concept combination should be annotated with an Appellation Event (is the term actually relevant). To decide these two questions, vectors are created that describe what words surround a term representing a concept. The vectors consist of words and how likely those words appear together with a specific concept. For example, such a vector could specify that if a term in the text represents the concept of a pony, the term will most likely be preceded by the words "barn" and "ride," and followed by the words "grass" and "barn."

---

[86] If two sequences of characters (or strings) are the same the Jaro-Winkler distance is 1. Two complete different strings have a score of 0.

For each concept, which has been retrieved as a possible meaning for a term in step two, a vector is created, called *standard vector*, so that each concept is characterized by a list of most likely surrounding words. Standard vectors are created based on already annotated texts. In addition, if a concept in Conceptpower links to a Wikipedia entry, the text of the linked Wikipedia article is used as well to create standard graphs. This ensures that if a concept was not used in an Appellation Event before, it might still be possible to create a standard vector.

Step Four: Candidate Vector Creation

Once a standard vector has been created for each possible meaning of a term (each retrieved concept), a vector is created for each candidate term extracted in step one. I will call these vectors *candidate vectors*. The candidate vectors contain the words surrounding a term; they describe the context of a term.

Step Five: Similarity Calculation

For each candidate vector, the similarity to the standard vectors can now be calculated. The similarity calculation takes into account how many of the words in a standard vector appear in a candidate vector, how similar a candidate term is to the concept of a standard vector, and in cases of terms that represent person names, if the concept is of type person. The calculated similarities range between 0 and 1.

Step Six: Appellation Event Creation

In this last step, the extraction component selects from all candidate terms the terms that have a high similarity to the standard vectors. So far, the similarity value does not have to be very high (greater than 0.3) to yield useful results. This can be explained by the fact that

usually there is only one meaning of a term used for annotating texts. For example, there are three definitions for the term "experiment," but only one of them was so far used in Appellation Events. However, it can be expected that with a growing base of annotated texts, the minimum similarity value will have to increase.

Results and Next Steps

An example output of the extraction component is shown in Appendix C. The results of the component so far are not 100% correct. However, the component returns Appellation Events that in many cases correctly assign a concept to a term. For instance, for the text shown in Appendix C, twenty-four Appellation Events were found; twenty of those were assigned the correct concept. It still needs to be studied, how the found Appellation Events compare to Appellation Events that would be created by hand by an annotator. Most of the people mentioned in the given text are found by the component and linked to the correct concepts in Conceptpower. It is likely that those people would also be annotated by a human annotator. Keywords like "development," "embryo," or "vertebrate" are annotated with the correct concepts as well. However, how many of those terms would be annotated by a researcher needs to be evaluated. In some cases, only parts of a word were found (for example "embryo" in "embryology"), or terms were found that an annotator might choose not to annotate as a term might not be relevant given its specific position in a text.

The next steps regarding the extraction component would be to develop the prototype into a service that can be used by Vogon to automatically create Appellation Events. Vogon could send texts to the service and create suggestions for Appellation Events based on the output of the service. A user could then accept or reject the suggestions created by Vogon. How much this process would facilitate the Vogon annotation workflow would

need to be studied more closely. It needs to be determined how well the extracted Appellation Events match a user's understanding of what terms are relevant in a text. If the matching rate is low, the question arises if that might hamper the annotation workflow.

Another research area is the extraction of Relation Events. Based on already created Relation Events, an extraction component could try to find relationships between Appellation Events in texts. The automatic extraction of Relation Events might not only alleviate the annotation process, but could also be used as a text exploration tool. For example, Vogon could utilize such a component for showing relationships between terms that are likely to appear in a text given existing Relation Events. These suggestions could inform a user's analysis of a new text or guide the reader of a text.

7.2.2 Network Analysis

Besides managing annotation projects, Quadriga's main purpose is to function as a network repository that stores all the networks from different projects that annotate texts using Vogon. Such a repository could enable scholars to discover unknown connections between people and place, or provide new tools to study certain concepts and how they change. In addition, Quadriga supports collaboration between researchers and allows projects to use data from other projects.

With the continuously growing number of stored networks, the question of how these networks can be analyzed arises. While standard network analysis techniques can be applied to Quadruple networks, there are certain characteristics that need to be taken into consideration. In this section, I will describe some ideas about what kind of tools Quadriga could provide to analyze stored networks and what factors should be taken into account for the development of these tools.

Comparing Networks

Quadriga stores networks for texts. In the Quadriga System, a text network consists of all the Appellation Events and Relation Events that were submitted by a user at the same time (see section 4.6 – System architecture). While most text networks are being created for different texts, it is possible to create two or more networks for the same text to find out how the interpretation of several scholars differ, or because a text was already annotated but focused on a different topic. For these cases, a tool could be developed that provides functionality to compare networks.

There are two main challenges for the development of a network comparison tool. First, algorithms have to be implemented that compare different networks. Second, a visualization module needs to be developed that allows a human user to compare networks. There are several methods to calculate the similarity of graphs or networks. Schenker et al. provide an overview of the most common techniques. For example, it can be calculated what the biggest subgraph of two or more graphs are (how many nodes and edges are the same), or how many modifications are necessary (for example, adding or deleting nodes or edges) to transform one graph into another [Schenker et al. 2005]. Adding a component to Quadriga that calculates the similarity of graphs could be used to provide a feature that suggests texts to a reader of a document based on the similarity of the document's network to other text graphs.

The development of a visualization component is required in cases in which users analyze texts or interpretation of texts and need a visual representation of the differences between two or more graphs. For example, if two scholars annotated the same text and would like to examine the differences, the two text networks need to be visualized in a way

that makes it easy to detect similarities and dissimilarities. Depending on the size of the two networks in question, a simple union of the two networks visualized using different colored nodes and edges (one color for each scholar and a third color for agreement) might be sufficient. However, if more than two networks are compared, this method might create visualizations that become too complex to be useful to a user. For such cases, methods that focus only on areas of agreement or disagreement might be more practical. For projects that focus on the changes in a network over time, to study for instance conceptual change, an animation or step-wise approach might be advisable.

Pattern Discovery

Text graphs are created with respect to the subject of a text and the research interest of a project. Several text graphs together define how the network of concepts used in the texts' annotations is structured. An interesting research question might be if there are specific patterns in these conceptual networks that can be found across different fields or times. For instance, in a network representing people, their institutions, and collaborations, can we detect a pattern that indicates a change in affiliation of a specific person such as increased collaboration with researchers from the other institution? Or similarly, before a new scientific concept is introduced, can we find a specific pattern of how the concepts related to that new concept are linked?

A component that tries to find such patterns would most likely make use of graph similarity techniques similar to the ones described above. There are several techniques to find patterns or structures in graphs (see for instance [Yan and Han 2006]). However, a pattern discovery component for Quadriga would have to take into account certain characteristics of the Quadriga System. For example, to detect a pattern for some nodes in a

Quadruple network, the type of a concept is more important than the concept itself, while for others the actual concept is significant. In a network of people and institutions it might be less relevant what specific person a node represents. However, it might be a distinct characteristic of a pattern that the person was a teacher of someone else, which would be represented by a teacher node and a relationship between the person and the teacher node.

Another difficulty for a pattern detection component would be that, although it is recommended to use standard graphs, every project can create its own standard graphs. This leads to several differently structured graphs that express the same or a similar fact. For example, a teacher-student relationship between two people could be expressed as person A is a student of person B, or person B is a teacher of person A. For accurately detecting patterns in Quadruple networks, the detection component would have to learn or be provided with information describing equivalence between different graph structures.

7.3 Infrastructure Enhancements

The Quadriga System is so far a closed system that, besides the document repository, uses only services that were specifically developed for Quadriga. To fully leverage its potential, additional resources could be connected to the Quadriga System and further software could be developed. This section describes some possible future developments that could integrate Quadriga into a larger infrastructure.

7.3.1 Semantic Search Engine

A logical enhancement of the Quadriga System would be the implementation of a search engine, which uses the Appellation Events stored in Quadriga rather than words. Such a search engine, in the following also called '*semantic search engine*, would allow users to not just

search for the occurrence of a specific term such as "Einstein," which would find all texts that contain the word "Einstein" regardless if the term denotes Albert Einstein, his son, or someone completely different, but to search for texts that actually mention that specific person (according to the annotator of a text).

Such a search engine could connect the document repository (or at some point potentially repositories) and Quadriga by querying Quadriga for Appellation Events using a particular concept, and returning texts from the document repository (see Figure 38). To realize this kind of semantic search engine, Quadriga would have to be extended to either return a list of Appellation Events or a list of texts that were annotated with those Appellation Events given a concept.

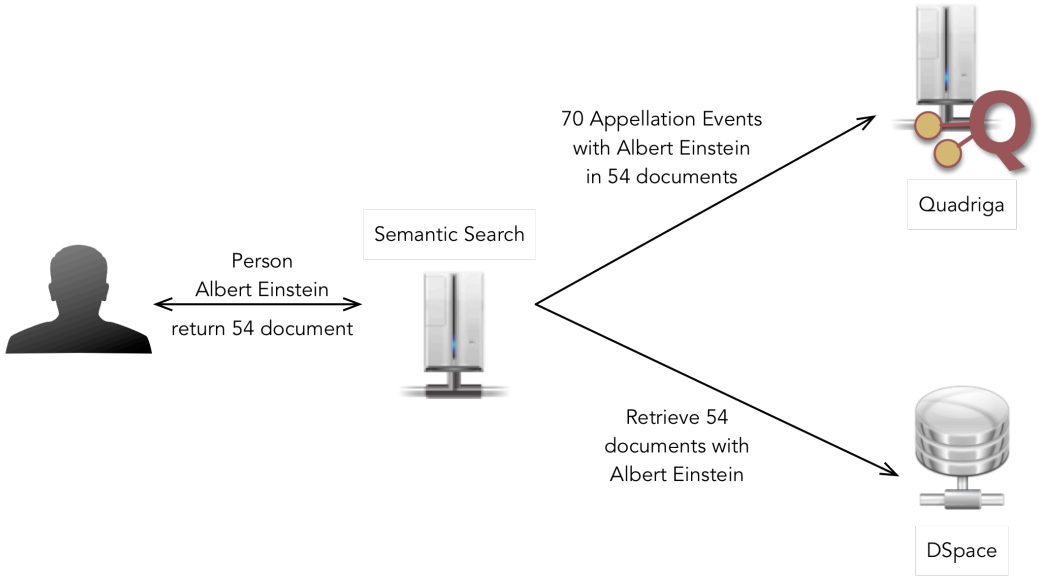

Figure 38: Semantic Search

The next level of development for this kind of search would be the development of a user interface that allows a user to search for relationships rather than concepts. For instance, a user could not only find all texts that talk about Albert Einstein, but all texts that

state that Albert Einstein was a physicist. This, however, presents several challenges. While searching for a particular concept is rather straightforward regarding user interface development (it could for example be an input field that supports the selection of a concept with an autocomplete function), developing an interface for creating relationship queries is more challenging. There are different approaches for user interface development such as drop-down lists for each subject, predicate, and object (as for instance described in [Auer et al. 2007]), or graphical query builder such as described in [Chau et al. 2008]. In addition, as mentioned before, each project might express the same or similar relationships using different Quadruple structures. A semantic search engine would therefore have to solve the problem of finding results that are not exact matches to a query but represent similar statements.

7.3.2 External Relationship Sources

There are many projects that produce data in triple format. The DBPedia project[87] mines all articles in Wikipedia and extracts relationships between and about the "things" described in the articles [Auer et al. 2007]. Others, such as the Wittgenstein Source project[88], create relationship data for the Semantic Web by hand [Pichler and Zöllner-Weber 2012]. The Quadriga System could benefit greatly from such external data source to inform network analysis and automatic relationship extraction processes, or to offer document exploration and search functionalities.

To include external data sources they would need to be translated for the Quadriga System to understand the external data. This mainly requires connecting the external data to

[87] http://dbpedia.org/
[88] http://www.wittgensteinsource.org/

Conceptpower. In cases in which external data sources refer to authority files such as VIAF or GeoNames, the translation process is rather easy, as many Conceptpower entries refer to those services as well. In the translation process concepts from Conceptpower simply have to be mapped to concepts in the external data source using VIAF or GeoNames URIs. However, if no common authority files are used, Quadriga can only "guess" what concepts are equivalent. A mapping file could be created that maps the types used in an external data source to the types used in Conceptpower. Quadriga could then try to find similarities between external and Conceptpower concepts regarding relationships to other entities. For instance, there could be a concept in Conceptpower named "Einstein" with the following relationships

Einstein - born - 1879
Einstein - married - Elsa Einstein

An external data source could use a concept called "Albert Einstein" with the relationships

Albert Einstein - is born - 1879
Einstein - married to - Elsa Einstein
Einstein - has son - Hans Albert Einstein

Because the concept names are similar as well as the relationships, a translation component could "guess" that the two concepts "Einstein" and "Albert Einstein" are the same. However, this example is highly simplified. A translation component that tries to automatically map concepts from different data sources will need further research.

7.3.3 A Service-based Infrastructure

The Quadriga-System is only one way of analyzing texts. It is a system that is based on the meso-level idea. Close reading is combined with collaborative projects and sharing and reuse of data to eventually enable the analysis of big datasets. There are other techniques (on

micro-, meso-, or macro-level) that can inform the analysis of sources, and the analysis processes of each other. Quadriga (representing the Quadriga System) could be part of a bigger infrastructure that allows a scholar to build pipelines of applications to analyze materials of interest. In this section I will describe a possible architecture for such a service-based text analysis infrastructure.[89,90]

The core component for an infrastructure that allows running several analysis tools on the same text corpus, is an application that functions as project management tool (in the following *management application*) and service registry. A service in this context is any kind of analysis technique offered as a web application that can be run on a corpus. The main functionality of the management application is the management of text corpora and available services, as well as a project management feature that allows users to run specific services on their text corpora.

The architecture I am proposing is based on loosely coupled components and allows registering any kind of service with the management application, as long as the service provides a specific kind of interface. This way the development management application stays mostly independent from changes in analysis applications, and new tools can be added at any time. To be able to register an analysis component with the management tools, the component has to offer a specific set of functions via a web API. This might be returning descriptive information about its analysis methods, and an interface to send the materials to be analyzed as well as retrieve the analysis results. To register a new service with the management application, the URL of the web API is stored in a list of available services.

---

[89] The basic idea for this kind of infrastructure was developed jointly by Manfred Laubichler's computational HPS group at Arizona State University and Colin Allen's lab at Indiana University. The architecture of the application proposed in this section was developed by me and Erick Peirson.

[90] See [Gold 2009] for a detailed description of service-based architectures, their history and benefits.

The infrastructure suggested here is *job*-based. To run a specific method on a source or corpus, a new job is created, which is sent to the service offering the method. The management application monitors the job and informs a user when the job is done. Depending on the service, the results can then either be downloaded and potentially added to a repository, or visualized by the responsible service for inspection by a user. The management application keeps a list of completed jobs and results with the option to use the results of one application as input for another tool. This way a *pipeline* of jobs can be created.

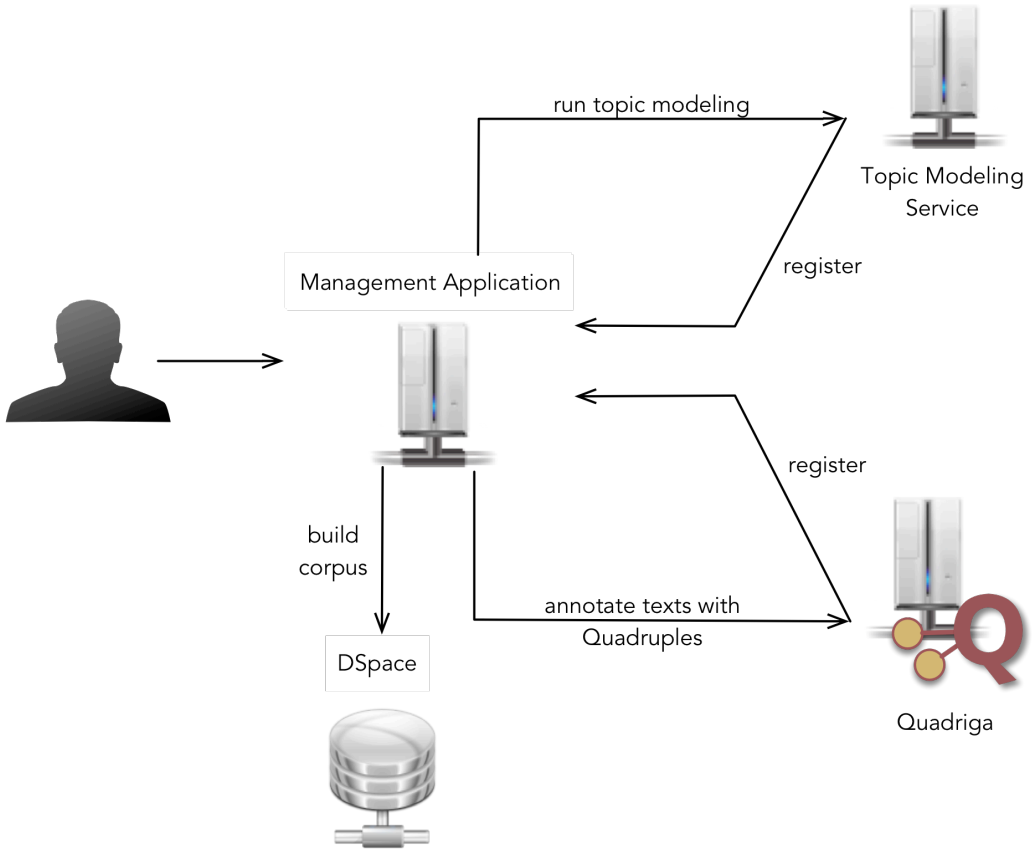

Figure 39: Service-based Infrastructure

The following example illustrates this architecture shown in Figure 39. There are two services registered with the management application: Quadriga and a topic modeling

application. A researcher wants to first run topic modeling on a corpus of texts, and then annotate a subset of these texts (all texts of one topic) using the Quadriga System. First, the researcher builds a corpus by selecting texts from the document repository. He then selects the topic modeling component from the list of available services in the management application, and runs it on his text corpus. Once the topic modeling process has finished, the researcher is notified and inspects the results. He selects all the texts of the topic of interest and chooses the Quadriga service to use next on these texts. The texts are sent to Quadriga, where the annotators of the researcher's project download them into Vogon and annotate them. Once the annotation networks for the selected texts are uploaded to Quadriga and approved, the researcher is again notified by the management application and redirected to Quadriga to study the resulting networks. If new text analysis services are registered with the management application, the researcher can use those on the existing text corpus. He might also choose to repeat the topic modeling process, or any other part of the analysis pipeline.

A service-based infrastructure as described in this section would make it possible to include any kind of text analysis (web) application. The only requirement would be that the application provides an interface according to the guidelines of the system. This would enable any project developing tools that they would like to connect to the infrastructure to add them to the pipeline. Such a system could increase reuse of software developed for digital HPS and might motivate the creation of (better) documentation and promotion of tools.

CHAPTER 8

CONCLUSION

> *"The Answer to the Great Question […] Of Life, the Universe and Everything […] Is …"*
> *"Yes…!!! …? "*
> *"Forty-two," said Deep Thought, with infinite majesty and calm.*
>
> — Adams [1979, p. 180-181]

In this dissertation I have described the design, development, and application of the Quadriga System. The system is designed to aid scholars in analyzing texts by transforming texts into structured datasets in the form of Quadruple networks that can be visualized and mathematically analyzed. However, I also examined the current landscape of digital HPS projects and described an approach to develop digital tools in a user-centric manner that benefits the education of computer science students as well as students in digital HPS/humanities.

8.1 Quadriga System

Quadruples, which are the underlying data structure of the Quadriga System, are contextualized triples of the form <subject - predicate - object - context>. Quadruples, in contrast to triples, store contextual information about a subject, predicate, object statement. Such contextual information contains, for instance, which text was annotated or who annotated a text. With this kind of data, it is possible to "zoom in and out" from the macro-level to the micro-level to allow scalable reading.

The Quadriga System operates on the meso-level, between distance and close reading. Texts are annotated through close reading and examination of terms. However, all annotations are stored in a common repository, creating a large-scale dataset of annotation

data (networks of Quadruples), which is available to other scholars. This dataset facilitates distant reading, which could assist in finding patterns and trends in the annotated corpus. However, distant reading in the context of the Quadriga System is limited by the number of annotated texts and the annotations created for those texts. Also, the annotation process itself is time-consuming and is likely to be restricted to a few texts of interest. Therefore, it might be practicable to use other distant reading techniques, such as topic modeling on large text corpora to identify sub-corpora of interest for ingestion into the Quadriga System. A project management software connecting to different kind of (text analysis) services could assist in that process by, for example, allowing a scholar to run a topic modeling tool on a corpus and from the results select a sub-corpus for annotation with Quadruples.

Compared to many large-scale text analysis methods, such as topic modeling or co-citation analysis, the Quadriga System has the advantage of not only connecting concepts and texts but also qualifying that link. For instance, a co-citation analysis might suggest that two papers are related because they are co-cited often, but it does not make any assertion about the kind of relationship between those papers. Do both papers make similar statements, or does one reject the statements of the other? Similarly, topic modeling might connect two terms by placing them in the same topic. However, it does no specify what kind of relationship exists between the two terms. Do texts that belong to a specific topic describe a similar relationship between these two terms, or are they using the same terms but contradict each other? The Quadriga System can answer such questions by qualifying the relationship between concepts. Two concepts are not only in relation to each other but are connected by a specific relationship. A scholar could use this property of the system by identifying several papers connected to each other by co-citation analysis, and then

annotating these papers with Quadruples to determine their specific relationships. In contrast to traditional close reading methods of the identified papers, the Quadriga System would provide a researcher not only with a structured way of extracting relevant information that can then be computationally analyzed using, for instance, network analysis measures. It would also allow a researcher to publish the extracted information (the Quadruple networks) so that other scholars can examine it or use it for their own research.

A crucial characteristic of the Quadriga System is its two-layer approach regarding interpretation. The "act of interpretation" of a text by a scholar plays a central role in the design of the system. Appellation Events and Relation Events capture information about when a researcher read and interpreted a text and how he understood it. Appellation Events carry information about the meaning of specific terms (the first layer of interpretation); Relation Events specify how the concepts (represented by terms and Appellation Events) in a text relate to each other according to an annotator (the second layer of interpretation). This approach impacts the system as a whole. Any network created in the Quadriga System has to be understood as an interpretation, and as the perspective of a particular scholar. Inconsistencies in networks are a critical feature of the system, as they can highlight debates or conflicting perspectives either in the literature or its interpretation.

Hyman and Renn state that "[e]xperimental data and historical sources are often reproduced only in a piecemeal fashion that does not allow for verification of the authors' conclusions without extensive research on one's own part" [Hyman and Renn 2012, p. 13-14]. Quadruples, with their feature to contain position information about what text passage exactly was interpreted, could be one solution to this problem. They turn text (which represents unstructured data to a computer) into structured data that can be (automatically)

compared, analyzed, and evaluated. By annotating texts with Quadruples representing the texts' interpretations, a scholar provides a basis for tracing back their argumentation to the textual source. Furthermore, arguments and interpretations can be compared by comparing the Quadruple networks. This might provide explanations for differences or similarities, such as different interpretations of single entities or their relations to each other.

Hyman and Renn envision for the Epistemic Web that it will bring together "knowledge from existing documents to represent new knowledge" [Hyman and Renn, p. 18]. By "federating documents" (either by hand or automatically) different sources of knowledge will be combined in order to provide access to several sources at once and to qualify the relationships of different sources [Hyman and Renn 2012]. The Quadriga System aims to achieve such a federation of knowledge by providing a central repository to store datasets from different projects. By using the same authority file, different projects can connect their data to the data of other projects. For instance, consider two projects focused on Charles Darwin. The first one annotates texts about his travels on the HMS Beagle. The second project annotates texts regarding his book "On the Origin of Species." If both projects use the same authority file (or authority files that are connect to each other), their annotations are connected and the Quadriga System "knows" that both projects are centered on Darwin. It could be possible then to trace back ideas such as the notion that variations that are advantageous for an organism are likely to be preserved throughout Darwin's texts. Especially in cases of research topics that are not well studied, this might reveal unknown or unexpected links between people, places, or theories. The Quadriga System offers a central location for data storage in a standardized way, so that data produced by separate projects

can be combined to enable researchers to ask questions that cannot be answered by one project alone.

Regarding the meaning of terms (called "concepts" in this dissertation), the Quadriga System does not try to define any terms in general besides a very broad description. The basic idea of the system is that a concept is defined by the concepts that it connects to. The Quadriga System embraces the fact that the meaning of terms and definition of concepts change over time, across disciplines, and people. Definitions are expected to emerge from the networks of Quadruples, which makes the system independent of a particular timespan or field of study, and could facilitate the detection and visualization of conceptual change.

8.2 Digital History and Philosophy of Science

The present landscape of digital HPS consists of a number of projects, which for the most part aim to present and disseminate data. There are only a few projects that use digital tools to support the analyzing of sources beyond making materials available online. The projects that are concerned with developing computational tools and methods for the most part do only a mediocre job of promoting and documenting the resulting software. Reuse of such computational tools seems to be rather low. I believe that improving this situation could be beneficial for a number of projects by providing new perspectives and modes of access to existing materials. Using methods that work on the macro-level could enable scholars to find new connections and patterns, or help novices in a field to explore a topic.

I believe that improving the current situation requires not only better documentation and promotion, but also the commitment to develop more generally applicable software tools. Specific research software needs to become generic research software. This can be achieved partially by developing service-oriented software, in which different services

provide independent functionalities. The Quadriga System follows this approach. Most of its components can also be used alone. For example, Conceptpower can be used as an authority file in any project, and Vogon can be used for creating and transforming networks independently of Quadriga. By designing the Quadriga System in this distributed way, potential reuse becomes more likely as other projects can "mix and match" as necessary.

A significant factor for promoting reuse of digital tools, however, is that software should be easy-to-use and well documented. The Digital Innovation Group can facilitate the process of creating user-friendly, documented, and reusable software by combining teaching of software engineering principles, training of future digital HPS/humanities scholars, with the creation and documentation of digital tools. By following an agile development methodology that fosters close collaboration between the developer and user of a piece of software, the digital tools are frequently reviewed by users who can provide feedback about usability and functionality. In addition, software cannot only be documented by its developers but also by its users, who are in some case much more qualified to write user manuals and tutorials.

In addition, the Digital Innovation Group enables students to acquire experience in cross-discipline collaboration, preparing them for their future careers. Students from (digital) HPS learn how to formulate their requests in a way that computer scientists can understand and translate into programming tasks. Computer science students gain hands-on experiences in how to satisfy the requirements of users, and how to communicate highly technical topics.

8.3 Onwards

In this dissertation, I have described current applications as well as possible enhancements for the Quadriga System. So far, I believe the Quadriga System is only in its beginnings with

several possible future research topics and enhancements. Some are specific to the Quadriga System, such as a component for automatic relationship extraction; some are concerned with the bigger picture and context in which the Quadriga System could fit.

Besides possible future development directions, however, it could also be valuable to apply the Quadriga System to other areas of research. For instance, Malt et al. studied how naming objects and classifying them by similarity compares across multiple languages. They found that although objects might be classified into different categories according to their names (for instance the term "chair" in English describes wooden as well as stuffed chairs, while in Chinese a large stuffed chair is referred to with the same term as a sofa), there seems to be no differences in perceived similarity [Malt et al. 1999].

The Quadriga System separates the terms used in a text from the concepts to which they refer. It therefore would be possible to annotate a text written in another language than English by using a dictionary for another language, but keeping the same authority file for referencing concepts. The authority file would have to be extended for concepts that are not present in English (for example, the German term *Schnapsidee*, which describes a stupid idea that foreseeably won't work), but many concepts would probably translate. A possible experiment could be to annotate texts about similar topics, but in different languages, and compare the resulting networks. Would these networks be similar to each other, or does using a different language entail different relationships between concepts? Another question could be if texts about a (scientific) theory but in different languages result in the same conceptual network? Ultimately, it would be an interesting research question to study if Quadruple networks could be used to aid automatic translation of texts. As they include contextual information about a specific term, such as time and author of the text that the

term appears in, a piece of software could use this additional information to translate texts even if translation and usage of a term might have changed.

A second possible research direction would be to extend the Quadriga System to allow the annotation of images with Quadruples. This would allow for a systematic comparison of interpretations of images across several scholars. Annotations could be placed on the image at the position where a specific concept appears or is expressed. Such information could also be used to guide a novice or generally interested viewer of a picture in how the elements in that image could be interpreted. At a minimum, both described research topics (multiple languages as well as image annotation) would enable the development of search tools that allow a user to find texts in different languages containing a specific concept or images depicting it.

The Quadriga System so far provides the basic infrastructure for Quadruple annotation projects. It provides tools for the creation, management, and visualization of Quadruple networks. However, the Quadriga System could develop into a much more powerful tool if embedded in a larger infrastructure that incorporates additional text analysis services as well as tools that sit on top of the Quadriga System using its data. The Quadriga System has also shown how education, research, and tool development can be brought together with the whole being greater than the sum of its parts.

REFERENCES

Ackerman, M. J. 1998. "The Visible Human Project." *Proceedings of the IEEE* 86 (3): 504-511.

Adams, Douglas. 1979. *The Hitchhiker's Guide to the Galaxy*. Del Ray.

ADHOa. "Alliance of Digital Humanities Organizations - About." Accessed January 26, 2014. http://adho.org/about.

ADHOb. "Roberto Busa Prize." http://adho.org/awards/roberto-busa-prize/. Accessed 22 January 2014.

Allemang, Dean and James Hendler. 2009. "RDF Store." In *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*, 64-73. Morgan Kaufmann. Accessed May 2, 2014. http://www.myilibrary.com?ID=127936.

American Museum of Natural History. "Picturing the Museum - About the Collection." Accessed November 7, 2013. http://images.library.amnh.org/photos/about.html.

Attwood, T. K., A. Gisel, N-E. Eriksson, and E. Bongcam-Rudloff. 2011. "Concepts, Historical Milestones and the Central Place of Bioinformatics in Modern Biology: A European Perspective." In *Bioinformatics - Trends and Methodologies*, 3-38. InTech.

Auer, Sören, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. "DBpedia: A Nucleus for a Web of Open Data." In *The Semantic Web, Lecture Notes in Computer Science*, 722–735. Springer Berlin Heidelberg.

Beck, Kent, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas, 2001. "Manifesto for Agile Software Development." Accessed April 9, 2014. http://agilemanifesto.org/.

Begel, Andrew and Nachiappan Nagappan. 2007. "Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study." *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, 255-264. Accessed July 22, 2014. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4343753.

Benjamins, V. R., J. Contreras, M Blázquez, J. M Dodero, A. Garcia, E. Navas, F. Hernandez, and C. Wert. 2004. "Cultural Heritage and the Semantic Web." In *The Semantic Web: Research and Applications*, edited by Christoph J. Bussler, John Davies, Dieter Fensel, and Rudi Studer, 433-444. Berlin, Heidelberg: Springer.

Benson, Dennis A., Ilene Karsch-Mizrachi, David J. Lipman, James Ostell, and Eric W. Sayers. 2010. "GenBank." *Nucleic Acids Research* 38 (Database issue): D46-51. Accessed July 22, 2014. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2808980/?tool=pmcentrez&report=abstract.

Berkeley Bioinformatics Open Source Project. 2013. "The Open Biological and Biomedical Ontologies." Accessed December 6, 2013. http://www.obofoundry.org/.

Berners-Lee, Tim. 2006. "Linked Data - The Story So Far." *Design Issues*. Accessed February 3, 2014; last update June 18, 2009. http://www.w3.org/DesignIssues/LinkedData.html.

Berners-Lee, Tim, Roy Thomas Fielding, and Larry Masinter. 2005. "Uniform Resource Identifier (URI): Generic Syntax." Accessed February 26, 2014. http://www.ietf.org/rfc/rfc3986.txt.

Berners-Lee, Tim, James Hendler, and Ora Lassila. 2001. "The Semantic Web." *Scientific American* 284 (5): 34-43.

Berry, David M. 2011. "The Computational Turn: Thinking About the Digital Humanities." *Culture Machine* 12 (The Digital Humanities: Beyond Computing). Accessed January 26, 2014. http://www.culturemachine.net/index.php/cm/issue/view/23.

Biodiversity Heritage Library. 2013. "Biodiversity Heritage Library." Accessed November 1, 2013. http://biodivlib.wikispaces.com/.

Bizer, Chris, Anja Jentzsch, and Richard Cyganiak. 2011. "State of the LOD Cloud." Accessed March 3, 2014; last updated September 19, 2011. http://lod-cloud.net/state/.

Bizer, Christian, Tom Heath, and Tim Berners-Lee. 2009. "Linked Data - The Story So Far." *International Journal on Semantic Web and Information Systems* 5 (3): 1-22. Accessed July 22, 2014. http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/jswis.2009081901.

Bloehdorn, Stephan, Peter Haase, York Sure, and Johanna Voelker. 2006. "Ontology Evolution." In *Semantic Web Technologies: Trends and Research in Ontology-based Systems*, edited by John Davies, Rudi Studer, and Paul Warren, 51-70. Hoboken, NJ: John Wiley & Sons.

Bodenreider, Olivier. 2004. "The Unified Medical Language System (UMLS): Integrating Biomedical Terminology." *Nucleic Acids Research* 32 (Database issue): D267-70. Accessed July 22, 2014. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC308795/?tool=pmcentrez&report=abstract.

Bordoni, Luciana. 2007. "Towards a Semantic Web: The Role of Ontologies in the Literary Domain." In *Futures Past: Thirty Years of Arts Computing*, edited by Anna Bentkowska-Kafel and Trish Cashen, 109-115. Bristol, GBR: Intellect Ltd. Accessed July 2, 2014. http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10161039&p00=10161039.

Brenan, J. P. M. 1966. "Noel Yvri Sandwith, 1901-1965." *Taxon* 15 (7): 245-255.

Brown, Kristen V. 2014. "Stanford's new Major Integrates Humanities, Computer Science." Accessed April 4, 2014. http://www.sfgate.com/technology/article/Stanford-s-new-major-integrates-humanities-5300471.php.

Brunak, Søren. 2012. "ELIXIR: The European Infrastructure for Biological Data and the Tools needed for their Analysis." Presentation at the 27th NORDUnet Conference at Oslo and Akershus University College on September 18, 2012. Accessed May 13, 2014. https://events.nordu.net/display/ndn2012web/ELIXIR%3A+The+European+infrastructure+for+biological+data+and+the+tools+needed+for+their+analysis+-+2.

Buckner, Cameron, Mathias Niepert, and Colin Allen. 2010. "From Encyclopedia to Ontology: Toward Dynamic Representation of the Discipline of Philosophy." *Synthese* 182 (2): 205-233.

Burrows, Toby. 2011. "Ontology Learning and the Humanities." In *Ontology Learning and Knowledge Discovery Using the Web*, edited by Wilson Wong, Wei Liu, and Mohammed Bennamoun, 186-99.

Busa, Roberto A. 2004. "Foreword: Perspectives on the Digital Humanities." In *A Companion to Digital Humanities*, edited by Susan Schreibman, Ray Siemens, and John Unsworth, xvi–xxi. Oxford: Blackwell Pub. Accessed January 26, 2014. http://www.digitalhumanities.org/companion/.

Caldarelli, Guido and Michele Catanzaro. 2012a. "A Fruitful Approach." In *Networks: A Very Short Introduction*, 7-22. Oxford University Press, Kindle edition.

Caldarelli, Guido and Michele Catanzaro. 2012b. "Digging Deeper into Networks." In *Networks: A Very Short Introduction*, 80-93. Oxford University Press, Kindle edition.

Cambridge University Library. 2013. "Board of Longitude." Accessed November 1, 2013. http://cudl.lib.cam.ac.uk/collections/longitude.

Cameron, Delroy, Pablo N. Mendes, Amit P. Sheth, and Victor Chan. 2010. "Semantics-Empowered Text Exploration for Knowledge Discovery." In *Proceedings of the 48th Annual Southeast Regional Conference on - ACM SE '10*. New York, NY: ACM Press.

Carroll, Jeremy J., Christian Bizer, Pat Hayes, and Patrick Stickler. 2005. "Named graphs, provenance and trust." In *Proceedings of the 14th international conference on World Wide Web - WWW '05*, 613-622.

Carroll, Jeremy J., and Patrick Stickler. 2004. "RDF Triples in XML." Technical report, HP Laboratories. Accessed May 2, 2014. http://www.hpl.hp.com/techreports/2003/HPL-2003-268.pdf.

Casties, Robert, and Martin Raspe. 2013. "Digilib - A versatile Image Viewing Environment for the Internet." Accessed December 5, 2013. http://digilib.berlios.de/.

Ceusters, Werner, and Barry Smith. 2011. "Switching Partners: Dancing with the Ontological Engineers." In *Switching Codes: Thinking Through Digital Technology in the Humanities and the Arts*, edited by Thomas Bartscherer and Roderick Coover, 103-124. Chicago, IL: University of Chicago Press.

Chau, Duen Horng, Christos Faloutsos, Hanghang Tong, Jason I. Hong, Brian Gallagher, and Tina Eliassi-Rad. 2008. "GRAPHITE: A Visual Query System for Large Graphs." In *ICDMW '08 Proceedings of the 2008 IEEE International Conference on Data Mining Workshops*, 963-966.

Chemical Heritage Foundation. 2010. "Rubber Matters." Accessed November 7, 2013. http://www.chemheritage.org/research/policy-center/oral-history-program/projects/rubber-matters/index.aspx.

Cock, Peter J A, Tiago Antao, Jeffrey T Chang, Brad a Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J L de Hoon. 2009. "Biopython: freely available Python tools for computational molecular biology and bioinformatics." *Bioinformatics (Oxford, England)* 25 (11): 1422-3. Accessed July 22, 2014. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2682512&tool=pmcentrez&rendertype=abstract.

Collen, Morris F. 1986. "Origins of Medical Informatics." *West J Med*, 145 (6): 778-785.

Colomb, Robert M. 2002. "Thesauri." In *Information Spaces: The Architecture of Cyberspace*, 160-162. Berlin, Heidelberg: Springer.

Crofts, Nick, Martin Doerr, Tony Gill, Stephen Stead, and Matthew Stiff. 2011. "Definition of the CIDOC Conceptual Reference Model, Version 5.0.4." Technical report, ICOM/CIDOC Documentation Standards Group and CIDOC CRM Special Interest Group.

Cyganiak, Richard and Anja Jentzsch. 2011. "The Linking Open Data Cloud Diagram." Accessed February 28, 2014; last updated September 19, 2011. http://lod-cloud.net/.

Cytoscape Consortium. 2013. "Cytoscape." Accessed January 18, 2014. http://www.cytoscape.org/.

Damerow, Julia. 2009. "Virtual Spaces MWN." Accessed November 27, 2013. http://virtualspaces.sourceforge.net/.

Damerow, Julia. 2013. "Arboreal MWN." Accessed November 27, 2013. http://arboreal.sourceforge.net/.

Bruijn, Jos de, Dieter Fensel, Mick Kerrigan, Uwe Keller, Holger Lausen, and James Scicluna. 2008. "Reasoning with WSML." In *Modeling Semantic Web Services: The Web Service Modeling Language*, 135-158. Berlin-Heidelberg: Springer.

De Nooy, Wouter, Andrej Mrvar, and Vladimir Batagelj. 2011. "Affiliations." In *Exploratory Social Network Analysis with Pajek*, 116-137. New York, NY: Cambridge University Press Textbooks.

Demuth, B., H. Hussmann, S. Zschaler, and L. Schmitz. 2000. "A Framework-based Approach to Teaching OOT: Aims, Implementation, and Experience." In *Thirteenth Conference on Software Engineering Education and Training*, 283-293. Accessed July 22, 2014. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=827055.

Deutsche Nationalbibliothek. "Katalog der Deutschen Nationalbibliothek." Accessed January 2, 2013. https://portal.dnb.de/opac.htm.

Deutsche Nationalbibliothek. 2013. "Integrated Authority File (GND)." Accessed January 2, 2014. http://www.dnb.de/EN/gnd.

Diggory, Mark and Bram Luyten. 2013. "Functional Overview." Accessed March10, 2014; last updated December 11, 2013. https://wiki.duraspace.org/display/DSDOC4x/Functional+Overview.

Digital HPS Consortium. 2013a. "Digital HPS Consortium." Accessed October, 31 2013. http://digitalhps.org/.

Digital HPS Consortium. 2013b. "Digital HPS Projects." Accessed October 31, 2013. http://digitalhps.org/projects.

Digital HPS Consortium. 2014a. "Henslow Letters Project." Accessed May 5, 2014. http://digitalhps.org/node/117.

Digital HPS Consortium. 2014b. "Poincaré Correspondence Project." Accessed May 5, 2014. http://digitalhps.org/node/22.

Doerr, Martin. 2003. "The CIDOC Conceptual Reference Module: An Ontological Approach to Semantic Interoperability of Metadata." *AI Magazine* 24 (3): 75-92.

Doerr, Martin. 2013. "CIDOC CRM Home page." Accessed December 13, 2013. http://www.cidoc-crm.org/.

Dumbill, Edd. 2003. "XML Watch: Tracking provenance of RDF data." Accessed October 3, 2013. Not accessible anymore. http://www.ibm.com/developerworks/xml/library/x-rdfprov/index.html.

EADH. "European Association for Digital Humanities - About." Accessed January 26, 2014. http://eadh.org/about.

Eck, Matthias, Stephan Vogel, and Alex Waibel. 2004. "Improving Statistical Machine Translation in the Medical Domain using the Unified Medical Language System." In *Proceedings of the 20th international conference on Computational Linguistics - COLING '04*, Morristown, NJ: Association for Computational Linguistics.

ELIXIR. 2014. "ELIXIR Structure." Accessed May 14, 2014. http://www.elixir-europe.org/about/elixir-structure.

Encyclopaedia Britannica Online. 2012. "s.v. 'Istanbul'." Accessed October 30, 2012. http://www.britannica.com/EBchecked/topic/296962/Istanbul.

Fan, Weiguo, Linda Wallace, Stephanie Rich, and Zhongju Zhang. 2006. "Tapping the Power of Text Mining." *Communications of the ACM* 49 (9): 76-82.

Fayad, Mohamed and Douglas C Schmidt. 1997. "Object-oriented Application Frameworks." *Communications of the ACM* 40 (10): 32-38.

Finin, Tim. 2001. "Reply to NAME: SWOL versus WOL." Accessed February 26. 2014. http://lists.w3.org/Archives/Public/www-webont-wg/2001Dec/0169.html

Fitzpatrick, Kathleen. 2012. "The Humanities, Done Digitally." In *Debates in the Digital Humanities*, edited by Matthew K. Gold, 12-15. Minneapolis, MN: University of Minnesota Press.

Freeman, Eric, Elisabeth Robson, Bert Bates, and Kathy Sierra. 2004. *Head First Design Patterns*. Sebastopol, CA: O'Reilly Media, Inc.

Gašević, Dragan, Dragan Djurić, and Vladan Devedžić. 2009. "Reasoning." In *Model Driven Engineering and Ontology Development*, 113-116. Springer.

Gibbs, Fred and Trevor Owens. 2012. "Building Better Digital Humanities Tools: Toward Broader Audiences and User-centered Designs." *Digital Humanities Quarterly* 6 (2). Accessed April 28, 2014. http://www.digitalhumanities.org/dhq/vol/6/2/000136/000136.html.

Gillies, Sean. 2012. "Pleiades Software Reuse." Accessed April 25, 2014. http://sgillies.net/blog/1137/pleiades-software-reuse/.

Gold, Nicolas. 2009. "Service-Oriented Software in the Humanities: A Software Engineering Perspective." *Digital Humanities Quarterly* 3 (4). Accessed October 30, 2012. http://www.digitalhumanities.org/dhq/vol/3/4/000072/000072.html.

Goldberg, Debra S. and Elizabeth K. White. 2014. "E Pluribus, Plurima: The Synergy of Interdisciplinary Class Groups." In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (SIGCSE '14) 457-462. New York, NY: ACM.

Groth, Paul, Andrew Gibson, and Jan Velterop. 2010. "The Anatomy of a Nanopublication." *Information Services and Use* 30: 51-56.

Gruber, Thomas R. 1993. "A Translation Approach to Portable Ontology Specifications." *Knowledge Acquisition* 5 (2): 199-220.

Gruber, Thomas R. 1995. "Toward principles for the design of ontologies used for knowledge sharing?" *International Journal of Human-Computer Studies* 43 (5-6): 907-928. Accessed October 30, 2012. http://www.sciencedirect.com.ezproxy1.lib.asu.edu/science/article/pii/S1071581985710816. doi: 10.1006/ijhc.1995.1081

Gruber, Tom. 2009. "Ontology." In *Encyclopedia of Database Systems*, 1963-1965. Accessed October 30, 2012. http://www.springerlink.com/content/r81025/#section=380154&page=1&locus=61.

Guha, Ramanathan, Rob McCool, and Richard Fikes. 2004. "Contexts for the Semantic Web." In *The Semantic Web – ISWC 2004*, Lecture Notes in Computer Science, 32-46. Berlin, Heidelberg: Springer.

Gupta, Anil. 2008. "Definitions." In *The Stanford Encyclopedia of Philosophy*, edited by Edward N Zalt. Fall 2012 Edition. Accessed July 22, 2014. http://plato.stanford.edu/archives/fall2012/entries/definitions/.

Gutierrez, Claudio, Carlos Hurtado, and Alejandro Vaisman. 2007. "Introducing Time into RDF." *IEEE Transactions on Knowledge and Data Engineering* 19 (2): 207-218.

Haase, Peter and Ljiljana Stojanovic. 2005. "Consistent Evolution of OWL Ontologies." In *The Semantic Web: Research and Applications*, edited by Asunción Gómez-Pérez and Jérôme Euzenat, 182-197. Berlin, Heidelberg: Springer.

Hahn, Walter v. and Cristina Vertan. 2006. First International Workshop Ontology Based Modelling in Humanities: 7 - 8 April 2006, University of Hamburg. Bibliothek des Department Informatik, Univ.

Haines, Catharine M. C. 2001. "Kennard, Olga née Weisz." In *International Women in Science: A Biographical Dictionary to 1950*, 157-159. ABC-CLIO.

Harms, Patrick and Jens Grabowski. 2011. "Usability of Generic Software in e-Research Infrastructures." *Journal of the Chicago Colloquium on Digital Humanities and Computer Science* 1 (3): 1-18.

Harvard University. 2004. Archimdes Project. Accessed October 31, 2013. http://archimedes.fas.harvard.edu/.

Hayles, N. Katherine. 2012. "How we think: Transforming Power and Digital Technologies." In *Understanding Digital Humanities*, edited by David M. Berry, 42-66. London: Palgrave Macmillan.

Heath, Tom and Christian Bizer. 2011. "Introduction." In *Linked Data: Evolving the Web into a Global Data Space*, edited James Hendler and Frank van Harmelen. Morgan & Claypool Publishers.

Hebeler, John, Matthew Fisher, Ryan Blace, and Andrew Perez-Lopez. 2009a. "Incorporating Semantics." In *Semantic Web Programming*, 93-140. Hoboken, NJ: Wiley-Blackwell. Accessed November 13, 2012. http://lib.myilibrary.com?ID=236887.

Hebeler, John, Matthew Fisher, Ryan Blace, and Andrew Perez-Lopez. 2009b. "Modeling Knowledge in the Real World." In *Semantic Web Programming*, 141-184. Hoboken, NJ: Wiley-Blackwell. Accessed November 16, 2012. http://www.myilibrary.com?ID=236887.

Hebeler, John, Matthew Fisher, Ryan Blace, and Andrew Perez-Lopez. 2009c. "Querying the Semantic Web." In *Semantic Web Programming*, 192. Hoboken, NJ: Wiley-Blackwell. Accessed November 16, 2012. http://lib.myilibrary.com?ID=236887.

Hellman, Eric. 2009. "Part 3: Reification Considered Harmful." Accessed March 13, 2014. http://go-to-hellman.blogspot.com/2009/05/part-3-reification-considered-harmful.html.

Henderson-Sellers, Brian. 2012. "Ontologies." In *On the Mathematics of Modelling, Metamodelling, Ontologies and Modelling Languages*, 47-54. Berlin, Heidelberg: Springer-Verlag. Accessed October 29, 2012. http://lib.myilibrary.com?ID=393973.

Henrich, Andreas and Tobias Gradl. 2013. "DARIAH(-DE): Digital Research Infrastructure for the Arts and Humanities – Concepts and Perspectives." *International Journal of Humanities and Arts Computing* 7 (1): 47-58.

Hersh, William R. 2002. "Medical Informatics - Improving Health Care Through Information." *JAMA* 288 (16): 1955-1958.

Hinsen, Konrad, Konstantin Läufer, and George K Thiruvathukal. 2009. "Essential Tools: Verison Control Systems." *Computing in Science & Engineering* 11 (6): 84-91.

Hockey, Susan. 2004. "The History of Humanities Computing." In *A Companion to Digital Humanities*, edited by Susan Schreibman, Ray Siemens, and John Unsworth. Hoboken, NJ: Blackwell Pub. Accessed January 26, 2014. http://www.digitalhumanities.org/companion/.

Hogeweg, Paulien. 2011. "The roots of bioinformatics in theoretical biology." *PLoS Computational Biology* 7 (3): e1002021. doi: 10.1371/journal.pcbi.1002021.

Horrocks, Ian. 2008. "Ontologies and the Semantic Web." *Communications of the ACM* 51 (12): 58-67.

Humphreys, Betsy L, Donald A. B. Lindberg, Harold M Schoolman, and G. Octo Barnett. 1998. "The Unified Medical Language System: An Informatics Research Collaboration." *J Am Med Inform Assoc* 5 (1): 1–11.

Hyman, Malcolm D. 2007. "Semantic Networks: A Tool for Investigating Conceptual Change and Knowledge Transfer in the History of Science." In *Übersetzung und Transformation*, edited by H. Böhme, C. Rapp, and W. Rösler, 355-367. Berlin: De Gruyter.

Hyman, Malcolm D and Jürgen Renn. 2012. "Toward an Epistemic Web." *Working Paper Series des Rates für Sozial- und Wirtschaftsdaten*, 197.

Information Society Technologies. 2004. "VICODI." Accessed December 13, 2013. http://www.vicodi.org/.

Jacobson, Daniel, Dan Woods, and Greg Brail. 2011. "The API Opportunity." In *APIs: A Strategy Guide*, 1-9. Sebastopol, CA: O'Reilly Media, Inc.

Kasneci, Gjergji, Fabian M. Suchanek, Georgiana Ifrim, Shady Elbassuoni, Maya Ramanath, and Gerhard Weikum. 2008. "NAGA: Harvesting, Searching and Ranking Knowledge." *SIGMOD '08 Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 1285-1288.

Killcoyne, Sarah, Gregory W Carter, Jennifer Smith, and John Boyle. 2009. "Cytoscape: A Community-Based Framework for Network Modeling." *Protein Networks and Pathway Analysis*, 563: 219-239.

Kintsch, Walter. 1998. "The Representation of Knowledge in Minds and Machines." *International Journal of Psychology*, 33 (6): 411–420.

Klein, Michel. 2001. "XML, RDF, and Relatives." *IEEE Intelligent Systems* 16 (2): 26-28.

Lawrence, Cera R. 2008. "Preformationism in the Enlightenment." Accessed March 30, 2014. http://embryo.asu.edu/handle/10776/1926.

LEAF Consortium. 2004. "Public Deliverables." Accessed January 2, 2014. http://www.leaf-eu.org/public.html.

LEAF Consortium. 2011. LEAF. Accessed January 2, 2014. http://www.leaf-eu.org/.

Lehmann, Fritz. 1992. "Semantic Networks." *Computers & Mathematics with Applications*, 23 (2-5): 1-50.

The Library of Congress. 2012. Library of Congress Authorities. Accessed January 2, 2014. http://authorities.loc.gov/.

The Library of Congress. 2013. Library of Congress Online Catalog. Accessed January 2, 2014. http://catalog.loc.gov/.

Liu, Chang. 2005. "Enriching Software Engineering Courses with Service-Learning Projects and the Open-Source Approach." In *Proceedings of the 27th International Conference on Software Engineering*, ICSE, 613-614, New York, NY: ACM.

Loesch, Martha Fallahay. 2011. "VIAF (The Virtual International Authority File) - http://viaf.org." *Technical Services Quarterly* 28 (2): 255-256. Accessed July 22, 2014. http://www.tandfonline.com/doi/abs/10.1080/07317131.2011.546304.

Lunenfeld, Peter, Anne Burdick, Johanna Drucker, Todd Presner, and Jeffrey Schnapp. 2012a. "A Short Guide to the Digital_Humanities." In *Digital_Humanities*, 121-136. Cambridge, MA: MIT Press.

Lunenfeld, Peter, Anne Burdick, Johanna Drucker, Todd Presner, and Jeffrey Schnapp, 2012b. "Distant/Close, Macro/Micro, Surface/Depth." In *Digital_Humanities*, 39-40. Cambridge, MA: MIT Press.

Macgregor, Robert and In-Young Ko. 2003. "Representing Contextualized Data using Semantic Web Tools." In *International Workshop on Practical and Scalable Semantic Systems PSSS1*.

Maienschein, Jane and Manfred D. Laubichler. 2009. "The Embryo Project: An Integrated Approach to History, Practices, and Social Contexts of Embryo Research." *Journal of the History of Biology* 43 (1): 1–16.

Maienschein, Jane and Manfred D Laubichler. 2012. "Charles Gillispie in the Digital Age." In *A Master of Science History*, edited by Jed Z. Buchwald, volume 30 of *Archimedes*, 37-45. Berlin, Heidelberg: Springer.

Malt, Barbara C, Steven A Sloman, Silvia Gennari, Meiyi Shi, and Yuan Wang. 1999. "Knowing versus Naming: Similarity and the Linguistic Categorization of Artifacts." *Journal of Memory and Language* 40: 230-262.

Maojo, Victor and Casimir A Kulikowski. 2003. "The Practice of Informatics: Viewpoint Paper: Bioinformatics and Medical Informatics: Collaborations on the Road to Genomic Medicine?" *J Am Med Inform Assoc* 10 (6): 515-523.

Martin, Chris. 2013. "Agnostic Editor." Accessed November 27, 2013. http://etcetera.caret.cam.ac.uk/blog/agnostic-editor.

Marx, Vivien. 2013. "The Big Challenges of Big Data." *Nature* 498: 255-260.

Mazza, Riccardo. 2009. "Introduction to Visual Representations." In *Introduction to Information Visualization*, 1-15. London: Springer. Accessed June 18, 2014. http://www.myilibrary.com?ID=203642.

McCarthy, John. 1980. "Circumscription–A Form of Non-Monotonic Reasoning." *Artificial Intelligence* 13 (1-2): 27-39.

McCarty, Willard. 2007. "Beyond retrieval? Computer Science and the Humanities." *Plenary Lecture for the CATCH Midterm Event*, Den Haag. Accessed April 5, 2014. http://www.mccarty.org.uk/essays/McCarty,%20Beyond%20retrieval.pdf.

Meeks, Elijah. 2011. "Digital Humanities as Thunderdome." *Journal of Digital Humanities* 1 (1). Accessed April 7, 2014. http://journalofdigitalhumanities.org/1-1/digital-humanities-as-thunderdome-by-elijah-meeks/.

Moody, Glyn. 2004. "The Code of Life." In *Digital Code of Life*, 1-9. Hoboken, NJ: John Wiley & Sons, Inc.

Moretti, Franco. 2000. "Conjectures on World Literature." *New Left Review*, 1: 54-68.

MPIWG. 2011. "The Musawwarat Graffiti Archive." Accessed November 7, 2013. http://musawwaratgraffiti.mpiwg-berlin.mpg.de/.

Müller, Christoph, Guido Reina, Michael Burch, and Daniel Weiskopf. 2012. "Large-Scale Visualization Projects for Teaching Software Engineering." *Computer Graphics and Applications, IEEE*, 14-19.

Müller, Martin. 2012. "Scalable Reading." Accessed April 25, 2014. https://scalablereading.northwestern.edu/scalable-reading/.

Müller-Birn, Claudia. 2014. "Informatik trifft Geisteswissenschaften." Interview in *Campus.leben*. Accessed April 5, 2014. http://www.fu-berlin.de/campusleben/forschen/2014/140225_digital-humanities/index.html.

Nagypal, Gábor, Richard Deswarte, and Jan Oosthoek. 2005. A"pplying the Semantic Web: The VICODI Experience in Creating Visual Contextualization for History." *Literary and Linguistic Computing* 20 (3): 327–349.

Naumann, Felix and Melanie Herschel. 2010. "Edit-Based Similarity." In *An Introduction to Duplicate Detection*, edited by M. Tamer Özsu, 30-34. Morgan & Claypool Publishers.

Navis, Adam R. 2007. "Samuel Randall Detwiler." Accessed April 21, 2014. http://embryo.asu.edu/handle/10776/1769.

Needleman, Mark H. 2001. "RDF: The Resource Description Framework." *Serials Review* 27 (1): 58–61.

Newman, Mark. 2010. "Bipartite Networks." In *Networks: An Introduction*, 123-126. Oxford University Press.

NLM. 1994. "The Visible Human Project - Getting the Data." Accessed June 30, 2014; last updated June 19, 2014. http://www.nlm.nih.gov/research/visible/getting_data.html.

NLM. 2003. "The Visible Human Project." Accessed June 30, 2014; last updated August 27, 2013. http://www.nlm.nih.gov/research/visible/visible_human.html.

NLM. 2011. "UMLS Quick Start Guide." Accessed June 30, 2014; last updated March 6, 2013. http://www.nlm.nih.gov/research/umls/quickstart.html.

Notess, Greg R. 2013. "Search Engine to Knowledge Engine?" *Online Searcher* 37 (4): 61-63.

OCLC. 2010. "Using the API." Accessed January 2, 2014. http://oclc.org/developer/documentation/virtual-international-authority-file-viaf/using-api.

OCLC. 2013a. "VIAF Contributor Institutions." Accessed January 2, 2014. http://www.oclc.org/viaf/contributors.en.html.

OCLC. 2013b. "VIAF Virtual International Authority File." Accessed October 28, 2013. http://www.oclc.org/viaf.en.html.

OCLC. 2014. "A brief history." Accessed July 20, 2014. https://oclc.org/viaf/history.en.html.

Opsahl, Tore. 2013a. "Triadic closure in two-mode networks: Redefining the global and local clustering coefficients." *Social Networks* 35 (2): 159-167.

Opsahl, Tore. 2013b. "Two-Mode Networks." Accessed March 13, 2014. http://toreopsahl.com/tnet/two-mode-networks/.

Oxford Dictionaries. 2010. "s.v. 'Concept'." Accessed October 29, 2012. http://oxforddictionaries.com/definition/american_english/concept.

Park, Jin Seo, Min Suk Chung, Sung Bae Hwang, Byeong-Seok Shin, and Hyung Seon Park. 2006. "Visible Korean Human: Its Techniques and Applications." *Clinical anatomy (New York, N.Y.)* 19 (3): 216-24.

Peirson, Erick, Julia Damerow, and Manfred Laubichler. 2014. "Don't Panic! A Research System for Network-based Digital History & Philosophy of Science." *Manuscript submitted for publication*. Pending.

Pichler, Alois and Amélie Zöllner-Weber. 2012. "Towards Wittgenstein on the Semantic Web." Accessed April 23, 2014. http://www.dh2012.uni-hamburg.de/conference/programme/abstracts/towards-wittgenstein-on-the-semantic-web/.

Powers, Shelley. 2003a. "Important Concepts from the W3C RDF Vocabulary/Schema." In *Practical RDF*, 83-99. Sebastopol, CA: O'Reilly Media, Inc.

Powers, Shelley. 2003b. "Ontologies: RDF Business Models." In *Practical RDF*, 228-252. Sebastopol, CA: O'Reilly Media, Inc.

Powers, Shelley. 2003c. "Specialized RDF Relationships: Reification, Containers, and Collections." In *Practical RDF*, 57-82. Sebastopol, CA: O'Reilly Media, Inc.

Powers, Shelley. 2003d. "The RDF Triple." In *Practical RDF*, 16-19. Sebastopol, CA: O'Reilly Media, Inc.

Presner, Todd. 2010. "Digital Humanities 2.0: A Report on Knowledge." In *The Connexions Project*, edited by Frederick Moody, Melissa Bailar, Ben Allen, Mary Ngolovoi, and Deborah Fay. Accessed January 29, 2014. http://cnx.org/content/m34246/1.6/.

Princeton University. 2013. "What is WordNet?" Accessed March 12, 2014; last updated November 7, 2013. http://wordnet.princeton.edu/wordnet/.

Prlić, Andreas, Andrew Yates, Spencer E. Bliven, Peter W. Rose, Julius Jacobsen, Peter V. Troshin, Mark Chapman, Jianjiong Gao, Chuan Hock Koh, Sylvain Foisy, Richard Holland,

Gediminas Rimsa, Michael L. Heuer, H. Brandstätter-Müller, Philip E. Bourne, and Scooter Willis. 2012. "BioJava: an open-source Framework or Bioinformatics in 2012." *Bioinformatics (Oxford, England)* 28 (20): 2693-5.

Reitz, Joan M. 2004a. "s.v. 'Authority Control'." In *Dictionary for Library and Information Science*, 53. Libraries Unlimited.

Reitz, Joan M. 2004b. "s.v. 'Authority Record'." In *Dictionary for Library and Information Science*, 53. Libraries Unlimited.

Reitz, Joan M. 2004c. "s.v. 'Controlled Vocabulary'." In *Dictionary for Library and Information Science*, 177. Libraries Unlimited.

Reitz, Joan M. 2004d. "s.v. 'Thesaurus'." In *Dictionary for Library and Information Science*, 716. Libraries Unlimited.

Renn, Jürgen, Jochen Büttner, Robert Casties, and Dirk Wintergrün. "A New Paradigm: the 'Digital Scrapbook'." Accessed December 5, 2013. http://www.mpiwg-berlin.mpg.de/en/research/projects/DEPT1_10_30Buettner-DigitalScrapbook.

Rodriguez, Marko A. 2011. "The RDF virtual machine." *Knowledge-Based Systems* 24 (6): 890-903.

Román, Jorge H., Kevin J Hulin, Linn M Collins, and James E Powell. 2012. "Entity Disambiguation Using Semantic Networks." *Journal of the American Society for Information Science and Technology* 63 (10): 2087-2099.

Roy Rosenzweig Center for History and New Media. 2013. "Omeka: Serious Web Publishing." Accessed December 5, 2013. http://omeka.org/about/

Sánchez, Diana Marcela, José María Cavero, and Esperanza Marcos Martínez. 2007. "The Road towards Ontologies." In *Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems. Integrated Series in Information Systems*, 3-20. Accessed October 30, 2012. http://lib.myilibrary.com.ezproxy1.lib.asu.edu/Open.aspx?id=81665.

Schenker, Adam, Horst Bunke, Mark Last, and Abraham Kandel. 2005. "Graph Similarity Techniques." In *Graph-theoretic Techniques for Web Content Mining*, 13-30. Singapore: World Scientific.

Schmidt, Desmond .2012. "The Role of Markup in the Digital Humanities." *Historical Social Research/Historische Sozialforschung* 3 (3): 125-146.

Schreibman, Susan. 2012. "Digital Humanities: Centres and Peripheries." *Historical Social Research/Historische Sozialforschung* 37 (3 (141)): 46-58.

Schreibman, Susan and Ann M. Hanlon. 2010. "Determining Value for Digital Humanities Tools: Report on a Survey of Tool Developers." *Digital Humanities Quarterly* 4 (2). Accessed April 28, 2014. http://digitalhumanities.org/dhq/vol/4/2/000083/000083.html#N100D3.

Schutz, Alexander. 2008. "XtraK4Me - Extraction of Keyphrases for Metadata Creation." Accessed April 21, 2014; last updated August 18, 2008. http://smile.deri.ie/projects/keyphrase-extraction.

Orozco, J. Martín Serrano. 2012. "Ontology Structures: Elements and Links." In *Applied Ontology Engineering in Cloud Services, Networks and Management Systems*, 56-58. Berlin, Heidelberg: Springer.

Seward, Sheraden. 2008. "James David Ebert (1921-2001)." Accessed March 29, 2014. http://embryo.asu.edu/handle/10776/1945.

Shannon, Paul, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. 2003. "Cytoscape: a software environment for integrated models of biomolecular interaction networks". *Genome research* 13 (11): 2498-504.

Shu, J., G. D. Clifford, W. J. Long, G. B. Moody, P. Szolovits, and R. G. Mark. 2004. "An Open-Source, Interactive Java-Based System for Rapid Encoding of Significant Events in the ICU Using the Unified Medical Language System." *Computers in Cardiology*, 197-200.

Siemens, Ray, Cara Leitch, Analisa Blake, Karin Armstrong, and John Willinsky. 2009. "'It May Change My Understanding of the Field': Understanding Reading Tools for Scholars and Professional Readers." *Digital Humanities Quarterly* 3 (4). Accessed October 30, 2012. http://www.digitalhumanities.org/dhq/vol/3/4/000075/000075.html.

Smith, Kaitlin. 2010. "William Keith Brooks." Accessed March 29, 2014. http://embryo.asu.edu/handle/10776/1683.

Sowa, John F. 1976. "Conceptual Graphs for a Data Base Interface." *IBM Journal of Research and Development* 20 (4): 336-357.

Sowa, John F. 2013. "Semantic Networks." Accessed May 3, 2014; last updated October 19, 2013. http://www.jfsowa.com/pubs/semnet.htm

Stanford NLP Group. 2014. "Stanford Named Entity Recognizer (NER)." Accessed April 21, 2014. http://nlp.stanford.edu/software/CRF-NER.shtml.

Starke, Gernot and Peter Hruschka. 2011. *Software-Architektur Kompakt: Angemessen Und Zielorientiert*. Berlin, Heidelberg: Springer.

Stoica, Marian, Marinela Mircea, and Bogdan Ghilic-Micu. 2013. "Software Development: Agile vs. Traditional." *Informatica Economica* 17 (4/2013): 64-76.

Svensson, Patrik. 2009. "Humanities Computing as Digital Humanities." *Digital Humanities Quarterly* 3 (3). Accessed January 22, 2014. http://digitalhumanities.org/dhq/vol/3/3/000065/000065.html.

Taylor, Arlene G. 1984. "Authority Files in Online Catalogs." *Cataloging & Classification Quarterly* 4 (3): 1–17.

Terras, Melissa. 2010. "DH2010 Plenary: Present, Not Voting: Digital Humanities in the Panopticon." Accessed October 18, 2012. http://melissaterras.blogspot.com/2010/07/dh2010-plenary-present-not-voting.html.

Terras, Melissa. 2012a. "Being the Other: Interdisciplinary Work in Computational Science and the Humanities." In *Collaborative Research in the Digital Humanities*, edited by Marilyn Deegan and Willard Mccarty, 226-243. Surrey, UK: Ashgate Publishing Group.

Terras, Melissa. 2012b. "On Making, Use and Reuse in Digital Humanities." Accessed April 25, 2014. http://melissaterras.blogspot.com/2012/03/on-making-use-and-reuse-in-digital.html.

The Eclipse Foundation. 2014. "Rich Client Platform." Accessed March 14, 2014; last updated December 2, 2013. http://wiki.eclipse.org/Rich_Client_Platform.

Tichy, Walter. 2010. "The Evidence for Design Patterns." In *Making Software: What Really Works, and Why We Believe It*, edited by Andy Oram and Greg Wilson, 393-414. Sebastopol, CA: O'Reilly Media, Inc.

Trilling, Bernie and Charles Fadel. 2009. "Digital Literacy Skills." In *21st Century Skills*, 61-71. Jossey-Bass.

Tuttle, Steven, Ami Ehlenberger, Ramakrishna Gorthi, Jay Leiserson, Richard Macbeth, Nathan Owen, Michael Storrs Sunil Ranahandola, and Chunhui Yang. 2006. "Directories." In *Understanding LDAP - Design and Implementation*, 5-10. IBM Redbooks.

University of Helsinki. 2014. "Anduril." Accessed January 19, 2014. http://www.anduril.org/.

University of Sheffield. 2014. "GATE." Accessed April 21, 2014. https://gate.ac.uk/.

U.S. National Library of Medicine. 2013a. "Detailed Indexing Statistics: 1965-2012." Accessed January 20, 2014; last updated March 22, 2013. http://www.nlm.nih.gov/bsd/index_stats_comp.html.

U.S. National Library of Medicine. 2013b. "Number of Titles Currently Indexed for Index Medicus® and MEDLINE® on PubMed®." Accessed January 20, 2014; last updated November 19, 2013. http://www.nlm.nih.gov/bsd/num_titles.html.

Veltman, Kim H. 2004. "Towards a Semantic Web for Culture." *Journal of Digital Information*, 4 (4).

W3C. 2013a. "SPARQL 1.1 Overview." Accessed April 18, 2014. http://www.w3.org/TR/sparql11-overview/.

W3C. 2013b. "SPARQL 1.1 Overview." Accessed March 12, 2014. http://www.w3.org/TR/sparql11-overview/.

W3C. 2013c. "SweoIG/TaskForces/CommunityProjects/LinkingOpenData." Accessed February 28, 2014; last updated November 26, 2013. http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData.

Waltzer, Luke. 2012. "Digital Humanities and the 'Ugly Stepchildren' of American Higher Education." In *Debates in the Digital Humanities*, edited by Matthew K. Gold, 335-349. Minneapolis, MN: University of Minnesota Press. Accessed May 15, 2014. http://site.ebrary.com/lib/asulib/Doc?id=10551807.

Weber, Jutta. 2004. "LEAF: Linking and Exploring Authority Files." *Cataloging & Classification Quarterly* 38 (3-4): 227-236.

Wikipedia Contributors. 2014a. "List of open-source bioinformatics software." In *Wikipedia*. Accessed January 18, 2014; last modified January 17, 2014. http://en.wikipedia.org/wiki/List_of_open-source_bioinformatics_software.

Wikipedia Contributors. 2014b. "Computer Science." Accessed April 7, 2014; last updated April 6, 2014. http://en.wikipedia.org/wiki/Computer_science.

Wikipedia Contributors. 2014c. "Software Engineering." Accessed April 7, 2014; last updated April 5, 2014.http://en.wikipedia.org/wiki/Software_engineering.

Wynne, Martin. 2013. "The Role of CLARIN in Digital Transformations in the Humanities." *International Journal of Humanities and Arts Computing* 7 (1-2): 89-104.

Yan, Xifeng and Jiawei Han. 2006. "Discovery of Frequent Substructures." In *Mining Graph Data*, 99-116. Hoboken, NJ: John Wiley & Sons, Inc.

Yu, Liyang. 2011. "Linked Open Data." In *A Developer's Guide to the Semantic Web*, 409-466. Berlin, Heidelberg: Springer.

Zhang, Na. 2013. "Using 'Gene Knock-Out' Techniques to Test Cultural Evolution." Accessed November 7, 2013. http://devo-evo.lab.asu.edu/cultural-evolution.

Zhang, Shao-xiang, Pheng-ann Heng, and Zheng-jin Liu. 2005. "Chinese visible human project: dataset acquisition and its primary applications." In *Engineering in Medicine and Biology Society, 2005 (IEEE-EMBS 2005)*, 4168-70.

APPENDIX A

THE PRESENT STATE

This chapter provides additional information to Chapter 2 – The Present State.

A.1 Digital HPS Projects

The following table lists all digital HPS projects that I evaluated in the context of my dissertation.

PROJECT	CATEGORY
Agnostic Editor	Computational Tools
Institutions: CARET http://etcetera.caret.cam.ac.uk/blog/agnostic-editor *Description*: The Agnostic Editor is being developed to simplify the editing process of structured data. It is optimized to work with XML texts. Its goal is eliminate the need to train editors of structured data such as XML text in "complex theory of structures." (Martin 2013) *Categorization Rationale*: The project is concerned with developing an XML editor tool and is therefore classified as "Computational Tool."	
Anteater	Computational Tools
Institutions: Max Planck Institute for the History of Science http://anteater-tool.sourceforge.net/ *Description*: Anteater is a text mining web application to extract certain information from Federal Register documents regarding endangered species research. *Categorization Rationale*: The project is concerned with developing a information extraction tool and is therefore classified as "Computational Tool."	
Arboreal	Computational Tools
Institutions: Max Planck Institute for the History of Science http://arboreal.sourceforge.net/ *Description*: "Arboreal MWN is a tool for content-based access to XML documents." (Damerow 2013) It provides functionality to read and edit XML documents, and to analyze those by using a language analysis service. *Categorization Rationale*: The project is concerned with developing an XML editing tool and is	

PROJECT	CATEGORY
therefore classified as "Computational Tool."	
Digilib	Computational Tools
Institutions: Max Planck Institute for the History of Science http://digilib.sourceforge.net/ *Description*: Digilib is a web application to view images in a web browser. "[D]igilib enables very detailed work on an image as required by scholars with elaborate viewing features like an option to show images on the screen in their original size." (Casties and Raspe 2013) *Categorization Rationale*: The project is concerned with developing an image viewer tool and is therefore classified as "Computational Tool."	
Digital Scrapbook	Computational Tools
Institutions: Max Planck Institute for the History of Science, Max Planck Digital Library http://www.mpiwg-berlin.mpg.de/en/research/projects/DEPT1_10_30Buettner-DigitalScrapbook *Description*: "The digital scrapbook supports the full spectrum of source-based scholarly work, from the first annotation to the final publication, in a unified format supported by the same tools." (Renn et al.) The Digital Scrapbook lets researchers share their collections, annotations, and references by making them accessible through the web. *Categorization Rationale*: The project is concerned with developing a virtual environment and tools to support humanistic research and is therefore classified as "Computational Tool."	
Omeka	Computational Tools
Institutions: George Mason University http://omeka.org/ *Description*: Omeka is a web-publishing platform to publish collections and virtual exhibitions of for example libraries or archives. "Omeka falls at a crossroads of Web Content Management, Collections Management, and Archival Digital Collections Systems." (Roy Rosenzweig Center for History and New Media 2013) *Categorization Rationale*: The project is concerned with developing a web-publishing application and is therefore classified as "Computational Tool."	
Virtual Spaces MWN	Computational Tools

PROJECT	CATEGORY
Institutions: Max Planck Institute for the History of Science http://virtualspaces.sourceforge.net/ *Description*: Virtual Spaces MWN is a tool to create virtual tours containing images, texts, and videos. These virtual tours can be exported and presented as webpages. (Damerow 2009) *Categorization Rationale*: The project is concerned with developing an application to develop virtual exhibitions and is therefore classified as "Computational Tool."	
Biodiversity Heritage Library (BHL)	Digital Collection
Institutions: Biodiversity Heritage Library Consortium consisting of a number of institutions such as American Museum of Natural History or Harvard University http://www.biodiversitylibrary.org/ *Description*: The goal of this project is to digitize literature on biodiversity and make it online available. *Categorization Rationale*: Although the BHL also aims to develop tools and services to support research that involves the material provided by BHL, its main purpose is to build up a digital collection of biodiversity literature (Biodiversity Heritage Library 2013, "Goal 1: Relevant Content").	
Board of Longitude	Digital Collection
Institutions: Cambridge University Library, National Maritime Museum, Board of Longitude Project http://blogs.rmg.co.uk/longitude/ http://cudl.lib.cam.ac.uk/collections/longitude *Description*: This project provides a digital version of all documents in the archives of the Royal Greenwich Observatory. In addition to the digitized materials, the project offers other sources such as educational materials and videos. *Categorization Rationale*: The project website of the Cambridge University Library states that "[t]his project [. . .] presents fully digitised versions of the complete archive and associated materials [. . .]" (Cambridge University Library 2013). This statement describes the core idea of a digital collection.	
Darwin Correspondence Project	Digital Collection
Institutions: University of Cambridge, Harvard University	

PROJECT	CATEGORY
http://www.darwinproject.ac.uk/ *Description*: The Darwin Correspondence Project provides transcripts of all letters Darwin wrote and received until 1869. Visitors of the website can read and search all these letters. *Categorization Rationale*: The main purpose of this project is to provide full text access to Darwin's letters. It therefore is classified as a "Digital Collection."	
Darwin Manuscripts Project	Digital Collection
Institutions: Drew University, American Museum of Natural History, Cambridge University Library, Natural History Museum, London, Biodiversity Heritage Library Consortium http://darwin.amnh.org/ *Description*: This project provides online access to Charles Darwin's scientific manuscripts. In addition to digital images of the manuscripts, the project provides transcripts for a part of the collection. *Categorization Rationale*: The project's main purpose is to provide online access to a collection of manuscripts. This is the definition for a "Digital Collection."	
Darwin-Hooker Letters	Digital Collection
Institutions: Cambridge Digital Library http://cudl.lib.cam.ac.uk/collections/darwinhooker *Description*: This projects provides online access to the digitized letters between Charles Darwin and Joseph Hooker. *Categorization Rationale*: The project's main purpose is the digitization and presentation of letters. It therefore falls into the category "Digital Collection."	
Einstein Papers Project	Digital Collection
Institutions: Hebrew University, California Institute of Technology http://www.alberteinstein.info/ *Description*: The purpose of the Einstein Papers Project is to provide online access to manuscripts by Albert Einstein. In addition to Einstein's scientific and non-scientific manuscripts, the project provides access to a database with information on the objects in the Albert Einstein Archives. *Categorization Rationale*: The project's main purpose is the digitization and presentation of manuscripts. It therefore falls into the category "Digital Collection."	

PROJECT	CATEGORY
European Cultural Heritage Online	Digital Collection
Institutions: Max Planck Institute for the History of Science, Bibliotheca Hertziana, and others http://echo.mpiwg-berlin.mpg.de/ *Description*: This project provides online access to digitized cultural heritage material in 95 collections covering several disciplines with a focus on the history of science. *Categorization Rationale*: The project's main purpose is the digitization and presentation of sources. It therefore falls into the category "Digital Collection."	
Henslow Letters Project	Digital Collection
Institutions: University of Cambridge, Harvard University http://www.darwinproject.ac.uk/ *Description*: "The Henslow Letters Project is a digital project that emerged during the process of com- piling and transcribing Charles Darwin's letters in the Darwin Correspondence Project. John Stevens Henslow (1796-1861) taught Charles Darwin and worked in the fields of botany and mineralogy. In order to access a sample of Henslow's full correspondences, the Darwin Correspondence Project can be currently queried to list those letters exchanged between Darwin and Henslow." (Digital HPS Consortium 2014a) *Categorization Rationale*: The project's main purpose is the digitization and presentation of letters. It therefore falls into the category "Digital Collection."	
Joseph Hooker Project	Digital Collection
Institutions: University of Sussex http://www.sussex.ac.uk/cweh/research/josephhooker *Description*: This project makes letters between Joseph Hooker and Indian colleagues (digital copies as well as their transcriptions) available in an online collection. *Categorization Rationale*: The project's main purpose is the digitization and presentation of letters. It therefore falls into the category "Digital Collection."	
Musawwarat Graffiti Archive	Digital Collection
Institutions: Max Planck Institute for the History of Science, Humboldt University Berlin http://musawwaratgraffiti.mpiwg-berlin.mpg.de/ *Description*: The Musawwarat Graffiti Archive provides an online collection of the graffiti on "the walls of the so-called Great Enclosure, a unique, labyrinthine building complex "	

PROJECT	CATEGORY
(MPIWG 2011). *Categorization Rationale*: The project's main purpose is to provide an online collection of pictures of Musawwarat Graffiti. It therefore falls into the category "Digital Collection."	
Newton Project	Digital Collection
Institutions: University of Sussex, Cambridge Digital Library http://www.newtonproject.sussex.ac.uk/prism.php?id=1 *Description*: The Newton Project provides online access to Newton's writings. His private and published documents have been transcribed and linked with translations, images, and other relevant information. *Categorization Rationale*: The project's main purpose is the digitization, presentation, and translation of Newton's manuscripts. It therefore falls into the category "Digital Collection."	
Picturing the Museum	Digital Collection
Institutions: American Museum of Natural History http://images.library.amnh.org/photos/index.html *Description*: This project provides an online collection of pictures from the museum following the tradition of Albert Bickmore, the founder of the museum, to "expand the Museum's educational mission beyond its walls." (American Museum of Natural History) *Categorization Rationale*: The project's main purpose is the digitization and presentation of images. It therefore falls into the category "Digital Collection."	
Poincaré Correspondence Project	Digital Collection
Institutions: Henri Poincaré Archives http://poincare.univ-lorraine.fr/ *Description*: "The Poincaré Correspondence Project seeks to transcribe the letters of French mathematician and philosopher, Henri Poincaré."(Digital HPS Consortium 2014b) *Categorization Rationale*: The project's main purpose is the digitization, transcription, and presentation of letters. It therefore falls into the category "Digital Collection."	
The Casebooks Project	Digital Collection
Institutions: University of Cambridge, University of Sussex http://www.magicandmedicine.hps.cam.ac.uk/	

PROJECT	CATEGORY
Description: The Casebooks Project provides a digital collection of approximately 80,000 medical records created by Simon Forman and Richard Napier between 1596 and 1634. *Categorization Rationale*: The project's main purpose is the digitization, transcription, and presentation of medical records. It therefore falls into the category "Digital Collection."	
The Cuneiform Digital Library Initiative	Digital Collection
Institutions: Max Planck Institute for the History of Science, University of California at Los Angeles http://cdli.mpiwg-berlin.mpg.de/ *Description*: This project provides online access to a catalogue of cuneiform tables with images of the tables and transliterations. *Categorization Rationale*: The project digitizes, transcribes, and presents of cuneiform tablets. It therefore falls into the category "Digital Collection."	
The Years of the Cupola	Digital Collection
Institutions: Max Planck Institute for the History of Science, Opera di Santa Maria del Fiore, Harvard University http://duomo.mpiwg-berlin.mpg.de/ *Description*: This project created a digital archive to provide online access to the documentary sources of the Opera di Santa Maria del Fiore. *Categorization Rationale*: The project digitizes, transcribes, and presents documents about the Opera di Santa Maria del Fiore. It therefore falls into the category "Digital Collection."	
Treasures of the Library	Digital Collection
Institutions: Cambridge Digital Library http://cudl.lib.cam.ac.uk/collections/treasures *Description*: This projects provides online access to a digital collection of the library in which documents are included that are considered to be "especially significant." *Categorization Rationale*: The purpose of this project is to create a "Digital Collection."	
University Of Oklahoma History Of Science Project	Digital Collection
Institutions: University of Oklahoma http://digital.libraries.ou.edu/homescience.php	

PROJECT	CATEGORY
Description: This project offers a number of online collections that provide access to books, pictures, and other source materials. *Categorization Rationale*: This project provides access to several "Digital Collections."	
Wallace Correspondence Project	Digital Collection
Institutions: Natural History Museum, London http://wallaceletters.info/ *Description*: The Wallace Correspondence Project has the goal to digitize and transcribe all existing letters from and to Alfred Russel Wallace. *Categorization Rationale*: The project digitizes, transcribes, and presents letters to and from Alfred Russel Wallace. It therefore falls into the category "Digital Collection."	
Archimedes Project	Digital Collection w/ Computational Tools
Institutions: Harvard University, Max Planck Institute for the History of Science, University of Missouri, Tufts University http://archimedes.fas.harvard.edu/ *Description*: The Archimedes Project provides a digital collection of sources in the history of mechanics. The collection is enhanced with language technologies that make the morphological analysis of texts, automatic indexing, and other text analysis features possible. *Categorization Rationale*: This project's basic service is to provide sources online. However, it combines the sources with computational functionality to aid researchers in analyzing the sources.	
Art of Life	Digital Collection w/ Computational Tools
Institutions: Missouri Botanical Garden, American Museum of Natural History, University of Florida, and others http://biodivlib.wikispaces.com/Art+of+Life *Description*: The Art of Life project aims to develop software to automatically identify and describe images in the Biodiversity Heritage Library. *Categorization Rationale*: This project's basic service is to provide sources online. However, it creates a collection by using computational methods.	
Chymistry Of Isaac Newton	Digital Collection w/ Computational Tools

PROJECT	CATEGORY

Institutions: Indiana University Libraries, Chemical Heritage Foundation

http://www.chymistry.org

Description: The Chymistry Of Isaac Newton project provides an online edition of Isaac Newton's alchemical manuscripts. It also provides a number of online tools for these texts such as a glossary or a tool analyze the relationships between terms, text parts, and documents in the Newton corpus.

Categorization Rationale: This project's basic service is to provide sources online. However, it combines the sources with computational functionality to aid researchers in analyzing the sources.

Cultures of Knowledge	Digital Collection w/ Computational Tools

Institutions: University of Oxford

http://www.culturesofknowledge.org/

Description: This project provides an online catalogue of letters sent in the 16th, 17th, and 18th century. In addition to the catalogue, the project aims to develop computational tools to analyze the data in the catalogue.

Categorization Rationale: This project's basic service is to provide sources online. However, it combines the sources with computational functionality to aid researchers in analyzing the sources.

Old Weather Project	Digital Collection w/ Computational Tools

Institutions: Met Office Hadley Centre, University of Oxford, National Oceanic and Atmospheric Administration, U.S. National Archives, National Maritime Museum

http://www.oldweather.org/

Description: This project makes digitized ship logs available online and has users transcribe them.

Categorization Rationale: As this project makes digitized material available online, it is classified as digital collection. The project also aims to collect data by having users transcribe digitized logs. It is therefore classified as "Digital Collection w/ Data Collection."

Embryo Project	Digital Collection/Education

Institutions: Center for Biology and Society (ASU), Max Planck Institute for the History of Science, Marine Biological Laboratory

PROJECT	CATEGORY
http://embryo.asu.edu/ *Description*: The Embryo Project publishes articles about the historical and social contexts of reproductive medicine, developmental biology, and embryology in an online encyclopedia. The articles are written by researchers and students. *Categorization Rationale*: This project makes articles and images centered around the history of embryology available online, it therefore is a digital encyclopedia. It combines the creation of the encyclopedia with an educational endeavor and is therefore classified as "Digital Collection/Education."	
Isis Current Bibliography Of History Of Science	Online Bibliography
Institutions: University of Oklahoma http://www.ou.edu/cas/hsci/isis/website/ *Description*: The projects provides an online version of the printed Isis Current Bibliography Of History Of Science. *Categorization Rationale*: This project is an online version of a printed bibliography and is therefore classified as "Online Bibliography."	
ESciDoc	Online Repository
Institutions: Max Planck Society, FIZ Karlsruhe https://www.escidoc.org/ *Description*: EsciDoc is a repository to publish and manage digital objects. *Categorization Rationale*: The purpose of this project is to provide online access to digital publications and is therefore classified as "Online Repository."	
HPS Repository	Online Repository
Institutions: Center for Biology and Society (ASU), Marine Biological Laboratory http://hpsrepository.asu.edu/ *Description*: The HPS repository stores digital objects documenting the history and philosophy of science and makes them available online. *Categorization Rationale*: The project provides online access to digital publications and is therefore classified as "Online Repository."	
Indiana Philosophy Ontology Project	Project using Computational Tools

PROJECT	CATEGORY
Institutions: Indiana University https://inpho.cogs.indiana.edu/ *Description*: This project uses the articles from several sources (such as the Stanford Encyclopedia of Philosophy) to create a "dynamic ontology" using data mining techniques. *Categorization Rationale*: This project is classified as a "Project using computational tools" because it uses computational tools to analyze data in order to create a dynamic ontology.	
Science under Scrutiny	Project using Computational Tools
Institutions: Max Planck Institute for the History of Science http://tinyurl.com/lrn6t5l *Description*: This project led by Etienne Benson uses archival research methods together with computational analysis techniques to study how science, ethics, and law intersect in the context of biodiversity. A software tool called "Anteater" was developed to mine U.S. regulatory documents regarding endangered species research. *Categorization Rationale*: This project is classified as "Project using computational tools" because it uses computational tools to analyze data on a big scale in combination with close reading methods.	
Using "Gene Knock-Out" Techniques to Test Cultural Evolution	Project using Computational Tools
Institutions: Center for Biology and Society (ASU) http://devo-evo.lab.asu.edu/cultural-evolution *Description*: This project employs topic modeling techniques to answer the question if there exist "functional units" in historic literature collections that significantly influence the vocabulary and its structure of the corpus. (Zhang 2013) *Categorization Rationale*: This project is classified as "Project using computational tools" because it uses computational tools to analyze text in order to draw conclusions about a corpus.	
Congo Expedition 1909-1915	Virtual Exhibition
Institutions: American Museum of Natural History http://diglib1.amnh.org/ *Description*: This project created a virtual exhibition about an expedition of Herbert Lang and	

PROJECT	CATEGORY
James Chapin to the Belgian Congo from 1909 to 1915. This virtual exhibition provides images, videos, and texts about the exhibition. *Categorization Rationale*: The purpose of this project is to create a "Virtual Exhibition."	
Critical Mass: A History of Mass Spectrometry	Virtual Exhibition
Institutions: Chemical Heritage Foundation http://www.chemheritage.org/research/policy-center/oral-history-program/projects/critical- mass/default.aspx *Description*: This project provides online access to texts, images, and videos about the history of mass spectrometry. The material is presented in form of a virtual exhibition. *Categorization Rationale*: The purpose of this project is to create a "Virtual Exhibition."	
Pratolino: The History of Science in a Garden	Virtual Exhibition
Institutions: Max Planck Institute for the History of Science, Ente Provincia of Florence http://pratolino.mpiwg-berlin.mpg.de/ *Description*: This project provides an online exhibition that compares the history Garden of Pratolino with the modern situation of the garden. *Categorization Rationale*: The purpose of this project is to create an online exhibition and is therefore classified as "Virtual Exhibition."	
Rubber Matters	Virtual Exhibition
Institutions: Chemical Heritage Foundation http://www.chemheritage.org/research/policy-center/oral-history-program/projects/rubber- matters/index.aspx *Description*: "This exhibit [...] tells the story of the U.S. Synthetic Rubber Program during World War II from the unique perspective of our oral history interviewees." (Chemical Heritage Foundation 2010) *Categorization Rationale*: The purpose of this project is to create an online exhibition and is therefore classified as "Virtual Exhibition."	
Biology of Aging	Website
Institutions: MBLWHOI Library The website does not exist anymore. It was http://biologyofaging.org/blog/.	

PROJECT	CATEGORY

Description: The project aimed to make information about aging available online through a website.

Categorization Rationale: The purpose of the project is to make information available through a website and is therefore classified as "Website."

APPENDIX B

A QUADRUPLE-BASED RESEARCH SYSTEM

This appendix provides supplementary information to Chapter 4 – A Quadruple-Based Research System.

B. 1 Implementation Details

In this section, screenshots and additional information about the different software applications are included.

B.1.1 Vogon

Figure 40 shows an example Standard Graph as it could look like when created in a diagram tool (as for example yEd[91]). Regular font indicates that a node has a specific concept attached to them. Italic font and square brackets indicate a node for which only a type is specified.

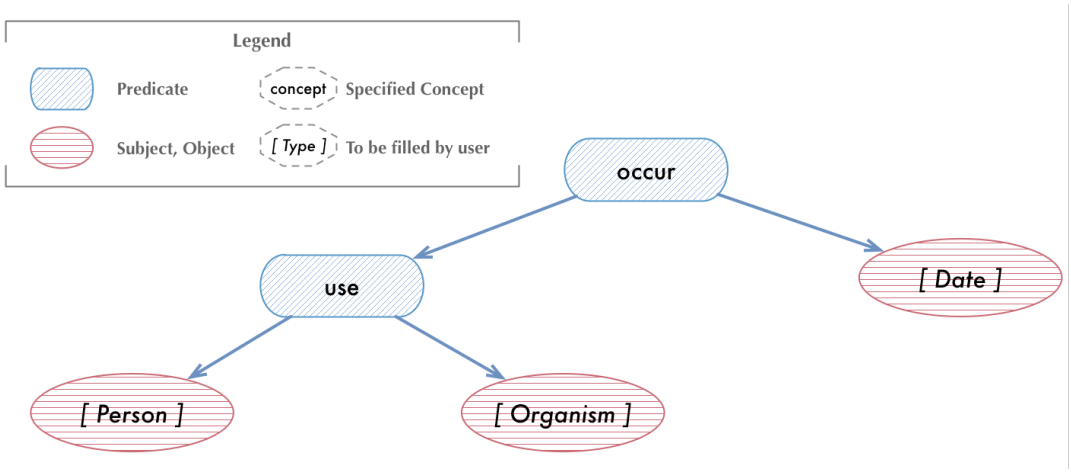

Figure 40: Example Standard Graph

Listing 10 shows an abbreviated version of the same Standard Graph as the one shown in Figure 40 in GraphML format.

[91] See http://www.yworks.com/en/products_yed_about.html

Listing 10: Example Standard Graph in GraphML

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns:y="http://www.yworks.com/xml/graphml"
          xmlns:yed="http://www.yworks.com/xml/yed/3"
          xsi:schemaLocation="http:// graphml.graphdrawing.org/xmlns
                  http://www.yworks.com/xml/schema/graphml/1.1/ ygraphml.xsd">
   <!--Created by yFiles for Java 2.10-->
   <key for="graphml" id="d0" yfiles.type="resources"/>
   [...]
   <graph edgedefault="directed" id="G">
       <data key="d10"/>
       <node id="n0">
         <data key="d4">
           <![CDATA[http://www.digitalhps.org/types/TYPE_986a7cc9 -c0c1 -4720-
b344-853f08c136ab]]>
         </data>
         <data key="d6"><![CDATA[ID1]]></data>
         <data key="d9">
            <y:ShapeNode>
              [...]
              <y:NodeLabel [...]>[ Person ] [...]</y:NodeLabel>
              <y:Shape type="ellipse"/>
            </y:ShapeNode>
         </data>
       </node>
       <node id="n1">
         <data key="d5">
           <![CDATA[http://www.digitalhps.org/concepts/WID-01158872-V -??-use]]>
         </data>
         <data key="d6"><![CDATA[ID2]]></data>
         <data key="d9">
           <y:ShapeNode>
             [...]
             <y:NodeLabel [...]>use [...]</y:NodeLabel>
             <y:Shape type="roundrectangle"/>
           </y:ShapeNode>
         </data>
       </node>
       <node id="n2">
          <data key="d4">
            <![CDATA[http://www.digitalhps.org/types/TYPE_01054126 -b6ec -4d31-
9b7f-7bc6738eb79a]]>
          </data>
          <data key="d6"><![CDATA[ID3]]></data>
          <data key="d9">
             <y:ShapeNode>
               [...]
               <y:NodeLabel [...] >[ Organism ] [...]</y:NodeLabel>
               <y:Shape type="ellipse"/>
             </y:ShapeNode>
          </data>
       </node>
       <node id="n3">
          <data key="d5">
```

```
54              <![CDATA[http://www.digitalhps.org/concepts/WID-00339934-V-??-
    occur]]>
55            </data>
56            <data key="d6"><![CDATA[ID4]]></data>
57            <data key="d9">
58               <y:ShapeNode>
59                  [...]
60                  <y:NodeLabel [...]>occur [...]</y:NodeLabel>
61                  <y:Shape type="roundrectangle"/>
62               </y:ShapeNode>
63            </data>
64         </node>
65         <node id="n4">
66            <data key="d4">
67               <![CDATA[http://www.digitalhps.org/types/TYPE_8beef031 -8f96 -440a-
    bd4e-0c3939536af1]]>
68            </data>
69            <data key="d6"><![CDATA[ID5]]></data>
70            <data key="d9">
71               <y:ShapeNode>
72                  [...]
73                  <y:NodeLabel [...] >[ Date ] [...]</y:NodeLabel>
74                  <y:Shape type="ellipse"/>
75               </y:ShapeNode>
76            </data>
77         </node>
78
79         <edge id="e0" source="n1" target="n2">
80            <data key="d11"><![CDATA[object]]></data>
81            <data key="d14">
82               <y:PolyLineEdge> [...] </y:PolyLineEdge>
83            </data>
84         </edge>
85         <edge id="e1" source="n3" target="n4">
86            <data key="d11"><![CDATA[object]]></data>
87            <data key="d14">
88               <y:PolyLineEdge> [...] </y:PolyLineEdge>
89            </data>
90         </edge>
91         <edge id="e2" source="n3" target="n1">
92            <data key="d11"><![CDATA[subject]]></data>
93            <data key="d13"/>
94            <data key="d14">
95               <y:PolyLineEdge> [...] </y:PolyLineEdge>
96            </data>
97         </edge>
98         <edge id="e3" source="n1" target="n0">
99            <data key="d11">
100              <![CDATA[subject]]>
101           </data>
102           <data key="d13"/>
103           <data key="d14">
104              <y:PolyLineEdge> [...] </y:PolyLineEdge>
105           </data>
106        </edge>
107     </graph>
```

```
    <data key="d0">
      <y:Resources/>
    </data>
</graphml>
```

B.1.2 Conceptpower

Figure 41 shows the webpage that can be used to search in Conceptpower.

Figure 42 shows the webpage for adding a new concept to Conceptpower.

Listing 11 shows a sample response of Conceptpower's web API.

Listing 11: Sample Conceptpower Response

```
<conceptpowerReply xmlns:digitalHPS="http://www.digitalhps.org/">
   <digitalHPS:conceptEntry>
      <digitalHPS:id concept_id="CON1c268257 -3f0e-4059-b4bb-a394ae2ce2a8"
          concept_uri="http://www.digitalhps.org/concepts/CON1c268257 -3f0e-4059-
b4bb-a394ae2ce2a8">
            http://www.digitalhps.org/concepts/CON1c268257 -3f0e-4059-b4bb-
a394ae2ce2a8
      </digitalHPS:id>
      <digitalHPS:lemma>Einstein</digitalHPS:lemma>
      <digitalHPS:pos> NOUN </digitalHPS:pos>
      <digitalHPS:description/>
      <digitalHPS:conceptList> Persons</digitalHPS:conceptList>
      <digitalHPS:creator_id>admin</digitalHPS:creator_id>
      <digitalHPS:equal_to>
         http://viaf.org/viaf/75121530
      </digitalHPS:equal_to>
      <digitalHPS:modified_by/>
      <digitalHPS:similar_to/>
      <digitalHPS:synonym_ids>
         WID-10954498-N-02-Albert_Einstein,
      </digitalHPS:synonym_ids>
      <digitalHPS:type
         type_id="986a7cc9-c0c1-4720-b344-853f08c136ab"
         type_uri="http://www.digitalhps.org/types/TYPE_986a7cc9-c0c1-4720-b344-
853f08c136ab">
           E21 Person
      </digitalHPS:type>
      <digitalHPS:deleted>false</digitalHPS:deleted>
      <digitalHPS:wordnet_id> WID-10954498-N-??-einstein</digitalHPS:wordnet_id>
  </digitalHPS:conceptEntry>
</conceptpowerReply>
```

Figure 41: Conceptpower Search

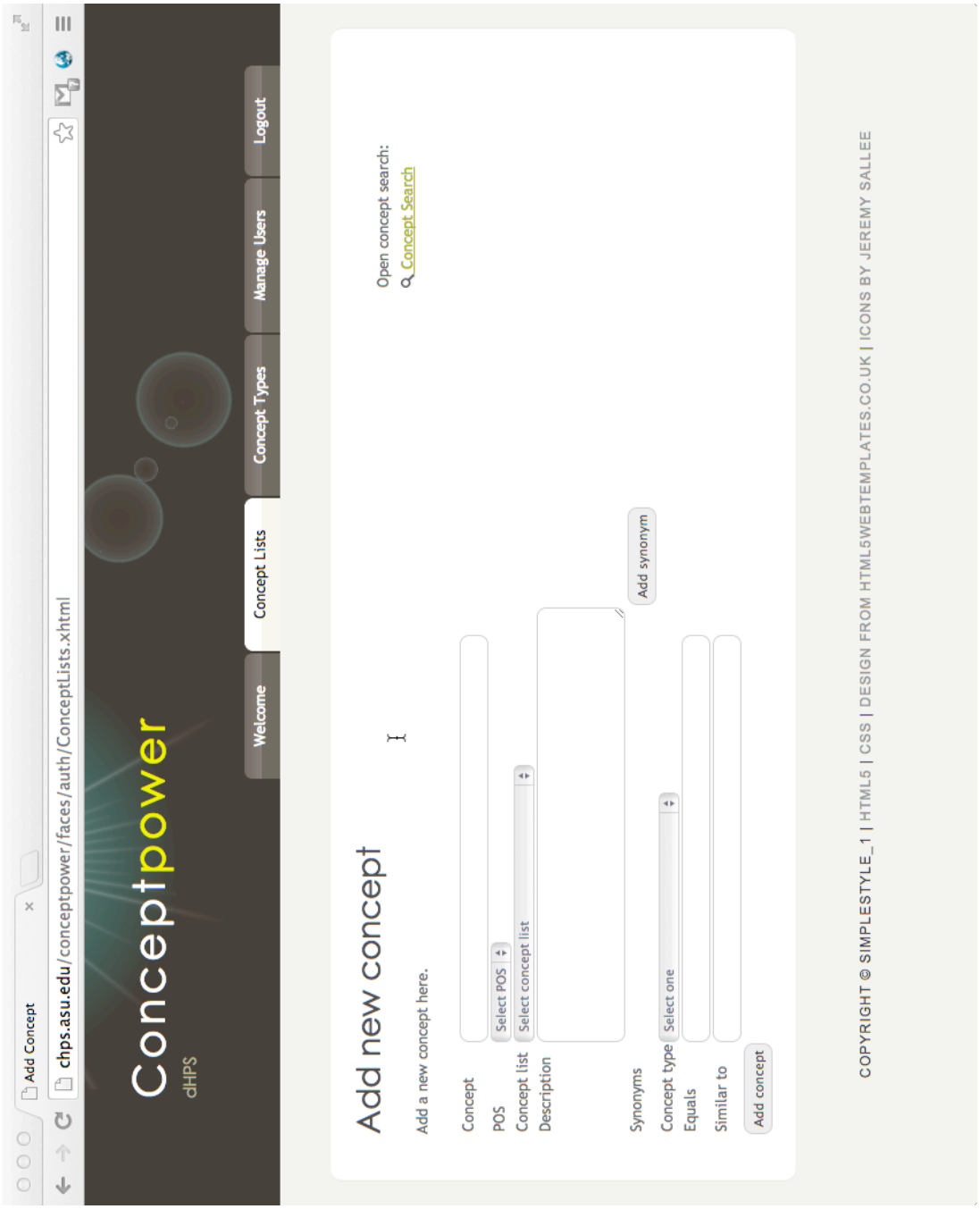

Figure 42: Add new Concept to Conceptpower

B.1.3 Wordpower

Figure 43 shows the webpage to search a term in Wordpower.

Figure 43: Search for a Term in Wordpower

Listing 12 shows a sample response of Wordpower's web API.

Listing 12: Sample Wordpower Response

```
<wordpowerReply xmlns:digitalHPS="http://www.digitalhps.org/">
  <digitalHPS:dictionaryEntry>
    <digitalHPS:id>
      http://www.digitalhps.org/dictionary/c53ef0a2 -8950-4035-a828-cc619ba31d34
    </digitalHPS:id>
    <digitalHPS:lemma>horse fetus</digitalHPS:lemma>
    <digitalHPS:pos>noun</digitalHPS:pos>
    <digitalHPS:description>fetus of a horse</digitalHPS:description>
    <digitalHPS:vocabulary>Biology Terms</digitalHPS:vocabulary>
  </digitalHPS:dictionaryEntry>
  <digitalHPS:dictionaryEntry>
    <digitalHPS:id>
     http://www.digitalhps.org/dictionary/XID-horse-n</digitalHPS:id>
    <digitalHPS:lemma>horse</digitalHPS:lemma>
    <digitalHPS:pos>noun</digitalHPS:pos>
    <digitalHPS:description>
        solid-hoofed herbivorous quadruped domesticated since prehistoric times a
        padded gymnastic apparatus on legs troops trained to fight on horseback; "500
        horse led the attack" a framework for holding wood that is being sawed a
        chessman shaped to resemble the head of a horse; can move two squares
        horizontally and one vertically (or vice versa)
    </digitalHPS:description>
    <digitalHPS:vocabulary>WordNet</digitalHPS:vocabulary>
  </digitalHPS:dictionaryEntry>
</wordpowerReply>
```

APPENDIX C

FUTURE WORK

This appendix contains supplementary information about possible future developments of the Quadriga System.

C.1 Automatic Term and Relationship Detection

The following excerpt from an Embryo Project article by Adam R. Navis was analyzed using various information extraction libraries [Navis 2007]. Due to processing power of the computer used, only the first four paragraphs of the original article were analyzed.

Samuel Randall Detwiler was an embryologist who studied neural development in embryos and vertebrate retinas. He discovered evidence for the relationship between somites and spinal ganglia, that transplanted limbs can be controlled by foreign ganglia, and the plasticity of ganglia in response to limb transplantations. He also extensively studied vertebrate retinas during and after embryonic development. Detwiler's work established many principles studied in later limb transplantation experiments and was identified by Viktor Hamburger as an important bridge between his and Ross Granville Harrison's research.

Detwiler was born on 17 February 1890 in Ironbridge, Pennsylvania, to Mary Hallman and Isaiah Detwiler. He was the youngest of twelve children and shared tasks on the family farm with his siblings. Detwiler was described as an energetic worker who rarely relaxed. Before attending college, he taught at the country schoolhouse near his family farm.

In 1910 Detwiler enrolled at Ursinus College in Collegeville, Pennsylvania, for two years. He completed his bachelor's degree at Yale University in 1914. Detwiler then began graduate work in zoology in Ross Harrison's laboratory. Detwiler earned a master's degree in 1916 and a PhD in anatomy and zoology from Yale in 1918. From 1917 to 1920 he maintained an appointment as an instructor at the Yale Medical School. In 1920 he accepted a position at the Peking Union Medical College in China where he spent three years. He then accepted a position as Assistant Professor of Zoology at Harvard University in

1923. He was promoted to Associate Professor in 1926 and travelled to Freiburg, Germany, to spend a semester in Hans Spemann's lab. In 1927 Detwiler became Professor of Anatomy at Columbia University.

Detwiler's research focused on neuroembryology and development of the vertebrate eye. He was a tenacious researcher. One series of experiments transplanting segments of spinal cord from one location of an embryo to another location of the embryo failed on the first one hundred attempts. His one hundred and first attempt was successful and produced a reliable technique. His work on neuroembryology began while he was working for Harrison. He published many papers concerning the development of vertebrate retinas, including a monograph summarizing his work. Detwiler performed many limb transplantation experiments. He discovered that transplanted limbs can be controlled by alternate ganglia. He also found the ganglia that received a transplanted limb grew larger and the ganglia that would have normally innervated the limb were smaller than usual. He found a direct relationship between the number of spinal ganglia and the number of somites in an embryo. Detwiler continued Harrison's work on neural development as Harrison moved into other fields.

The following listing shows a sample output of the Appellation Event extraction component for the above text.

Listing 13: Appellation Event Extraction Component Output

```
<text>
    Samuel Randall Detwiler was an
    <annotation term="embryo"
        conceptURI="http://www.digitalhps.org/concepts/CON65b0dba4-3029-4993-
8a44-da62a9507ca9"
        description="An embryo is a multicellular diploid eukaryote in its earliest stage
of development, from the time of first cell division until birth, hatching, or germination."
        confidence="0.5349617216914">embryo</annotation>
    logist who studied neural development in embryos and
    <annotation term="vertebrate"
        conceptURI="http://www.digitalhps.org/concepts/WID-01471682-N-??-
vertebrate"
        description="animals having a bony or cartilaginous skeleton with a segmented
spinal column and a large brain enclosed in a skull or cranium"
    confidence="0.5349617216914">vertebrate</annotation>
    retinas. He discovered evidence for the relationship between somites
    and spinal ganglia, that transplanted limbs can be controlled by
    foreign ganglia, and the plasticity of ganglia in response to limb
```

transplantations. He also extensively studied
<annotation term="vertebrate"
conceptURI="http://www.digitalhps.org/concepts/WID-01471682-N-??-vertebrate"
description="animals having a bony or cartilaginous skeleton with a segmented spinal column and a large brain enclosed in a skull or cranium"
confidence="0.3980314901687">vertebrate</annotation>
retinas during and after
<annotation term="embryo"
conceptURI="http://www.digitalhps.org/concepts/CON65b0dba4-3029-4993-8a44-da62a9507ca9"
description="An embryo is a multicellular diploid eukaryote in its earliest stage of development, from the time of first cell division until birth, hatching, or germination."
confidence="0.3980314901687">embryo</annotation>nic
<annotation term="development"
conceptURI="http://www.digitalhps.org/concepts/WID-13464820-N-??-development"
description="a process in which something passes by degrees to a different stage (especially a more advanced or mature stage);
the development of his ideas took many years the evolution of Greek civilization the slow development of her skill as a writer"
confidence="0.5349617216914">development</annotation>. Detwiler's work established many principles studied in later limb
transplantation experiments and was identified by
<annotation term="Viktor Hamburger"
conceptURI="http://www.digitalhps.org/concepts/CON8b731086-9479-4f4e-9995-c0ba92b2ab68"
description="The embryologist." confidence="0.9412514158049">Viktor Hamburger</annotation> as an important bridge between his and
<annotation term="Ross Granville Harrison"
conceptURI="http://www.digitalhps.org/concepts/CONac06020b-c357-4d7f-8286-3b07d07c3775"
description="(January 13, 1870 - September 30, 1959) American Biologist, Anatomist"
confidence="0.5349617216914">Ross Granville Harrison</annotation>'s research.

Detwiler was born on 17 February 1890 in Ironbridge, Pennsylvania, to Mary Hallman and Isaiah Detwiler. He was the youngest of twelve children and shared tasks on the family farm with his siblings. Detwiler was described as an energetic worker who rarely relaxed. Before attending college, he taught at the country schoolhouse near his family farm.

In 1910 Detwiler enrolled at Ursinus College in Collegeville, Pennsylvania, for two years. He completed his bachelor's degree at Yale University in 1914. Detwiler then began graduate work in zoology in
<annotation term="Ross Granville Harrison"
conceptURI="http://www.digitalhps.org/concepts/CONac06020b-c357-4d7f-8286-3b07d07c3775"
description="(January 13, 1870 - September 30, 1959) American Biologist, Anatomist"
confidence="0.5024300022032">Ross Harrison</annotation>'s laboratory. Detwiler earned a master's degree in 1916 and a PhD in
<annotation term="anatomy"
conceptURI="http://www.digitalhps.org/concepts/WID-06057539-N-??-anatomy"

description="the branch of morphology that deals with the structure of animals"
confidence="0.3980314901687">anatomy</annotation>
and zoology from Yale in 1918. From
<annotation term="1917"
conceptURI="http://www.digitalhps.org/concepts/CON493cd2d0-a7b4-4d28-b5cc-0098c0972eab"
description="1917 AD" confidence="0.3980314901687">1917</annotation>
to
<annotation term="1920"
conceptURI="http://www.digitalhps.org/concepts/CONdd1a6a72-34e3-41cc-9c82-4981f622bed8"
description="1920 AD" confidence="0.3980314901687">1920</annotation>
he maintained an appointment as an instructor at the Yale Medical
School. In
<annotation term="1920"
conceptURI="http://www.digitalhps.org/concepts/CONdd1a6a72-34e3-41cc-9c82-4981f622bed8"
description="1920 AD" confidence="0.3980314901687">1920</annotation>
he accepted a position at the Peking Union Medical College in China
where he spent three years. He then accepted a position as Assistant
<annotation term="Professor of Zoology"
conceptURI="http://www.digitalhps.org/concepts/CONb221f808-f7a8-47d0-84b4-e63f37c42f83"
description="Professor of Zoology" confidence="0.3980314901687">Professor of Zoology</annotation>
at Harvard University in 1923. He was promoted to Associate Professor
in
<annotation term="1926"
conceptURI="http://www.digitalhps.org/concepts/CON1fcff1fb-a796-483f-8c8e-8d3c46c19ce7"
description="1926 AD" confidence="0.3980314901687">1926</annotation>
and travelled to Freiburg, Germany, to spend a semester in
<annotation term="Hans Spemann"
conceptURI="http://www.digitalhps.org/concepts/CONf953d13b-dbb6-4f75-aae1-c507ba903f08"
description="the embryologist" confidence="0.9818073475236">Hans Spemann</annotation>'s lab. In
<annotation term="1927"
conceptURI="http://www.digitalhps.org/concepts/CONa0364401-a39b-4160-bd34-1d64feb3abb8"
description="1927" confidence="0.3980314901687">1927</annotation>
Detwiler became Professor of Anatomy at Columbia University.

Detwiler's research focused on neuro
<annotation term="embryo"
conceptURI="http://www.digitalhps.org/concepts/CON65b0dba4-3029-4993-8a44-da62a9507ca9"
description="An embryo is a multicellular diploid eukaryote in its earliest stage of development, from the time of first cell division until birth, hatching, or germination."
confidence="0.5349617216914">embryo</annotation>logy and development of the
<annotation term="vertebrate"
conceptURI="http://www.digitalhps.org/concepts/WID-01471682-N-??-vertebrate"
description="animals having a bony or cartilaginous skeleton with a segmented spinal column and a large brain enclosed in a skull or cranium"

confidence="0.3980314901687">vertebrate</annotation>
eye. He was a tenacious researcher. One series of experiments transplanting segments of spinal cord from one location of an embryo to another location of the
<annotation term="embryo"
conceptURI="http://www.digitalhps.org/concepts/CON65b0dba4-3029-4993-8a44-da62a9507ca9"
description="An embryo is a multicellular diploid eukaryote in its earliest stage of development, from the time of first cell division until birth, hatching, or germination."
confidence="0.5349617216914">embryo</annotation>
failed on the first one hundred attempts. His one hundred and first attempt was successful and produced a reliable technique. His work on neuro
<annotation term="embryo"
conceptURI="http://www.digitalhps.org/concepts/CON65b0dba4-3029-4993-8a44-da62a9507ca9"
description="An embryo is a multicellular diploid eukaryote in its earliest stage of development, from the time of first cell division until birth, hatching, or germination."
confidence="0.3980314901687">embryo</annotation>logy began while he was working for Harrison. He published many papers concerning the
<annotation term="development"
conceptURI="http://www.digitalhps.org/concepts/WID-13464820-N-??-development"
description="a process in which something passes by degrees to a different stage (especially a more advanced or mature stage);
the development of his ideas took many years the evolution of Greek civilization the slow development of her skill as a writer"
confidence="0.7480275777009">development</annotation>
of
<annotation term="vertebrate"
conceptURI="http://www.digitalhps.org/concepts/WID-01471682-N-??-vertebrate"
description="animals having a bony or cartilaginous skeleton with a segmented spinal column and a large brain enclosed in a skull or cranium"
confidence="0.7480275777009">vertebrate</annotation>
retinas, including a monograph summarizing his work. Detwiler performed many limb transplantation experiments. He discovered that transplanted limbs can be controlled by alternate ganglia. He also found the ganglia that received a transplanted limb grew larger and the ganglia that would have normally innervated the limb were smaller than usual. He found a direct relationship between the number of spinal ganglia and the number of somites in an
<annotation term="embryo"
conceptURI="http://www.digitalhps.org/concepts/CON65b0dba4-3029-4993-8a44-da62a9507ca9"
description="An embryo is a multicellular diploid eukaryote in its earliest stage of development, from the time of first cell division until birth, hatching, or germination."
confidence="0.3980314901687">embryo</annotation>.
Detwiler continued Harrison's work on neural
<annotation term="development"
conceptURI="http://www.digitalhps.org/concepts/WID-13464820-N-??-development"
description="a process in which something passes by degrees to a different stage (especially a more advanced or mature stage);
the development of his ideas took many years the evolution of Greek

```
        civilization the slow development of her skill as a writer"
        confidence="0.5349617216914">development</annotation>
    as Harrison moved into other fields.
</text>
```