

Enabling the Distributed Family Tree

by

Hilton Campbell

A thesis proposal submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science

Brigham Young University

November 2006

Abstract

While there is a large amount of genealogical data available on the Internet today, most of it is only found in isolated pockets. This causes genealogists difficulty when they try to locate pertinent information, and is often the cause of duplicated research efforts and paralyzing dead-ends. The objective of this thesis is to help overcome these encumbrances through the creation of a universally shared family tree that is distributed, open, scalable, extensible, standards-based, and machine-understandable. We will accomplish this through the specification of a data model and protocol for communications; the development of conformant server and client software; and the use of a natural language search interface, real-time data extraction of Web content, and semi-automatic lineage linkage to drive content expansion.

1 Introduction

1.1 Background and Motivation

The Internet offers a large selection of resources for the modern genealogist, such as: private and public databases of records and research results (e.g. [www.ancestry.com]); family websites with pedigree charts, anecdotes, and photos (e.g. [www.tribalpages.com]); genealogy wikis for collaboration (e.g. [www.rodovid.org]); and much, much more. To give a rough indication of the breadth of information available, Cyndi's List, the premiere directory of genealogical websites, lists over a quarter of a million verified and categorized links [www.cyndislist.com]. Likewise, WeRelate.org, a specialized search engine for Internet genealogy, boasts an index of over six million web pages from more than 1.3 million unique sources [www.werelate.org].

Unfortunately, most of this information is found in isolated pockets scattered across the Internet, which makes it difficult to locate specific records. This is problematic when researchers need to find that one key piece of information on the World Wide Web which would open up new avenues of research. The task is often like looking for the proverbial needle in a haystack. Frequently the genealogy researcher will end up duplicating prior work, or stall unnecessarily at a dead-end, even though the relevant information is available *somewhere* online.

These encumbrances are problems that all genealogists face, whether they are novices or professionals. All genealogists want better access to the information that is available, and they always want more information available. In light of these desires it is not difficult to see why the Holy Grail of computer-assisted genealogy is to bring together as much genealogical information as possible into one universally shared family tree [Qua03].

The creation of a universal family tree will make it possible to discover this otherwise elusive information, as well as provide compelling advantages such as the following.

- Novice genealogists will, with a single search, potentially be able to find all the information that has ever been compiled and published about their family on the Internet.
- As genealogists work on the same family tree, any information that is added will be made available to the others in real-time. Consequently, they may often receive unexpected assistance in their research through “accidental collaboration” from others.

- Genealogists will be able to search all published information on the human family tree through a single versatile interface.
- It will be easy to find other researchers working on the same family lines.
- Genealogists will be able to access their data almost any place and at any time because it is available online.

There are many other advantages to the creation of a universal family tree as well, and the benefits will only increase due to the network effects that will emerge when large numbers of researchers work in tandem.

In order for a universal family tree to provide these benefits, it must satisfy five key requirements.

- *Machine-understandable*: Software agents should be able to navigate the tree and perform research and other labor-intensive tasks on behalf of their human masters.
- *Open*: Researchers should be able to independently publish information without surrendering control of it or their intellectual property rights.
- *Standards-based*: Compliance with industry standards will accelerate adoption, make data exchange easier, and enable software reuse.
- *Extensible*: It is impossible to predict in advance all the kinds of genealogical information that researchers will eventually want to record.
- *Scalable*: The system must continue to function satisfactorily as the amount of information and usage increases.

By meeting these requirements, a universal family tree will be able to provide a significantly better level of service than the traditional World Wide Web in making it possible to find, share, and manipulate genealogical information.

1.2 The Distributed Family Tree

The *Distributed Family Tree (DFT)* is an open network of genealogical data and metadata which satisfies the above requirements. In essence, this network is nothing more than a collection of nodes, or servers on the Internet, each of which makes available some subset of the human family tree. Nodes are connected by relationships between the individuals described. Because the network is open, anyone can add new nodes and publish additional, even conflicting, information.

Anyone can add to the network without restraint, so the validity of published information is naturally suspect. This is an issue which plagues the World Wide Web at large and does not admit to any easy solution; it can, however, be mitigated. All genealogical information in the DFT has a corresponding provenance trail which genealogists can examine to help them ascertain the information's validity. Genealogists can designate what information and sources they believe are trustworthy, which customizes their views of the universal family tree and informs others. In

this way researchers can publish contradictory facts based on competing evidence without overriding what others accept as true.

Of course, this vision is not without its obstacles. Most genealogical data is in human-readable form only and tends to be very disconnected. Popular keyword search interfaces are inadequate for highly structured data, while the more appropriate advanced search interfaces tend to be inefficient and difficult to use. In general, people are slow to adopt new and unfamiliar technologies. This thesis will explore these problems and a handful of promising solutions in an attempt to stimulate the creation of the DFT.

1.2.1 Chicken-and-Egg Dilemma

The DFT suffers acutely from a chicken-and-egg dilemma. An existing network of genealogical information is needed before people can create applications to take advantage of it. Yet useful applications that demonstrate the power of the DFT are needed before people will go through the trouble of converting their own data. Like the World Wide Web, which suffered a similar dilemma at its inception, it will be necessary to bootstrap the DFT.

A real-time extraction technique developed by the Data Extraction Group at BYU can supplement the nascent DFT with much of the genealogical information already available in human-readable form on the World Wide Web. This technique uses specialized conceptual models called *extraction ontologies* to extract data from Web pages [Wal04].

Extraction ontologies are highly resilient to differences in page format, and can be used on a wide range of resources, because the conceptual model relies on relationships, lexical appearance, and contextual keywords, as opposed to page structure which tends to be brittle and does not generalize well. At the cost of greater computational effort and less than perfect accuracy, real-time data extraction make it possible for content originally authored for human consumption to automatically form a part of the DFT.

1.2.2 Inadequate Search Interfaces

A large amount of data such as will be found in the DFT is practically useless unless it can be efficiently searched. Simple keyword search, as popularized by Web search engines such as Yahoo! [www.yahoo.com] and Google [www.google.com], works well on a corpus of unstructured data but performs poorly on highly structured, finer grained information such as genealogical records. Advanced search interfaces can provide the granularity necessary to effectively search this data, but tend to be difficult and inefficient to use.

Drawing on the advantages of both simple keyword search and advanced search interfaces, the Data Extraction Group at BYU has developed a technique to translate natural language queries into domain-specific, machine-understandable queries [Vic06]. This technique uses extraction ontologies to take a simple query written in plain English or any other human language and translate it into an advanced query useful for searching structured data. Use of this technique can help genealogists to take full advantage of the DFT without any extensive training.

1.2.3 Isolated Pedigrees

Genealogical data usually consists of individual records and partial pedigrees. These records and pedigrees exist independently and seldom reference each other explicitly. Pedigrees must therefore be stitched together to construct a universal human tree, a task which will require significant time and human energy if done by hand.

Fortunately, computer software can assist in this effort. The Data Mining Lab at BYU has developed a technique to perform automatic lineage linkage [Pix06]. This technique compares pairs of individual records with a neural network to find duplicates. When the same individual exists in two different pedigrees, those pedigrees can be linked together. In this way, isolated pedigrees merge to form increasingly larger sub-trees of the human family tree.

This process cannot be completely automated, however, because the lineage linkage technique is not perfectly accurate. Rather than automatically create linkages between similar individuals, software that implements this technique would as a rule only make recommendations. The human user can then decide which recommendations are valid, which are not, and which cannot yet be determined.

Though not fully automated, this technique can dramatically increase the rate at which connections are made in the DFT. This is important because the overall utility of the network is directly proportional to its *connectivity* (the probability of any piece of genealogical information being connected to any other). Simulation studies on random directed and undirected graphs indicate that the degree of connectivity will slowly increase as connections are made, until the network reaches a critical point known as the *double jump*, when the network undergoes a phase transition and the overall connectivity rises dramatically [See00].

1.2.4 Adoption

Like any other technology, the greatest obstacle to the success of the DFT is adoption. It will thrive only if embraced by the genealogical community at large. The principle of network effects suggests that the utility of the network will increase exponentially as the number of participants increases. We hope, therefore, that the open and extensible nature of the DFT will motivate strong community interest and support.

2 Related Work

If the establishment of a universal family tree is the Holy Grail of computer-assisted genealogy, it would only make sense that there have been many attempts to achieve it. The following subsections will consider what others have done to make this happen and why it is not sufficient.

2.1 Global Genealogy Network

The *Global Genealogy Network* is a distributed network of genealogical information that was proposed but never completed because of resource constraints [JL01]. It consists of four conceptual layers.

- *Source Catalog*: A distributed compilation of digitized source documents, such as photos, scanned documents, audio/video clips, GEDCOM and PAF files, Web pages, microfiche, and so on.
- *Fact Catalog*: Files in a dialect of XML called GML (*Genealogy Markup Language*) which provide semantic metadata about the facts and sources contained in the source catalog.
- *Individuals and Family Relations Catalog*: An advanced search engine that uses an index and cache to search for genealogical data.

- *Parallelized Auto-completion*: Automated computer indexing of information and lineage linkage with the use of a distributed network.

In the DFT scheme, the source catalog is simply the Web, the fact catalog maps to the DFT network itself, and the last two layers are implemented in client software.

2.2 Peer-to-Peer Genealogy

The *Genealogy Network Transfer Protocol (GNTP)* is an unfinished protocol for a peer-to-peer genealogy network that was not completed because of resource constraints [ADJ+01]. The idea was to share GEDCOM files in much the same way that music and other files are distributed on other peer-to-peer networks.

In order to leverage already existing infrastructure, and because of its simplicity, the DFT will initially use a client/server model to publish, search for, and retrieve genealogical information. A directory of DFT nodes will be published on the Internet which client software can download and use to access one-by-one. This does not preclude the use of a peer-to-peer network, however. In fact, future work may extend the DFT with a peer-to-peer architecture to improve scalability and speed.

2.3 Real-Time Collaboration

In 2001, Dr. Scott Woodfield proposed the creation of a peer-to-peer virtual database [Woo01]. Conceptually, each user would belong to a different collaboration group for each individual of interest. New information on any given individual would be broadcast to all the other members of the group and then updated in each recipient's local database. In order to overcome the conceptual differences that exist between different researchers, each user would have a mediated view of the underlying database, allowing for conflicting claims. The DFT is a partial implementation of this proposal.

2.4 Genealogy Wikis

Several genealogy wikis have appeared recently, including WikiTree [www.wikitree.org] and Rodovid.org [www.rodovid.org]. One of the principle advantages of wikis is the lack of imposed structure on data. This proves to be a disadvantage in the case of genealogy, which is fundamentally structured. Nevertheless, it is possible to instrument wiki software so that it emits semantic metadata as well as human-readable data and thus forms part of the DFT.

2.5 Other Universal Family Trees

There have also been some attempts to create a universal family tree with a *centralized* approach.

- *OneGreatFamily.com*: A service which allows users to upload their genealogical data, performs lineage linkage, and notifies the user whenever new connections are made [www.onegreatfamily.com].
- *FamilySearch.org*: The family history department of the *Church of Jesus Christ of Latter-day Saints* is currently beta testing a new universal family tree which supports conflicting claims and views [www.familysearch.org].
- *Ancestry World Tree*: A collection of user submitted family trees which contains nearly 400 million names [www.ancestry.com/trees/awt].

While it is possible to create a centralized repository of genealogical data that everyone shares, there are compelling reasons to distribute that data instead. If genealogical information is scattered and replicated across multiple physical locations, the overall system will exhibit greater fault tolerance, availability, and scalability. More importantly, there is no need to resolve the intellectual property and control problems at the system level, as anyone with a website can publish genealogical information without submitting that information to a central database.

2.6 GENTECH Genealogical Data Model

The *GENTECH Genealogical Data Model* is a comprehensive logical data model for genealogical research and analysis, first published in 1998 [Lex00]. It was developed by the GENTECH Lexicon Group over a period of four years, with support from a number of genealogical societies, including the Federation of Genealogical Societies (FGS), the New England Historic Genealogical Society (NEHGS), the National Genealogical Society (NGS), the American Society of Genealogists (ASG), the Association of Professional Genealogists (APG), and the Board for Certification of Genealogists (BCG). The purpose of the data model is to “define genealogical data and the relationships between that data in an effort to bring greater understanding to the genealogical community about data issues” [Lex00].

The GENTECH data model is a meticulous description of the genealogical process and product. Indeed, many argue that the model is too meticulous to be usable, citing the lack of software that implements it for evidence. While this may hold true under the relational model assumed by the GENTECH data model, the problem becomes significantly more tractable with the use of a graph-based model, as is the case with the DFT. For this reason, the design of the DFT data model will be strongly influenced by concepts and relationships defined in the GENTECH data model.

3 Thesis Statement

The foundation for the *Distributed Family Tree (DFT)*, an open network of genealogical data and metadata which is scalable, extensible, standards-based, and machine-understandable, can be established through the following.

- A specification for a graph-based data model which can record genealogical, provenance, and trust information will be produced.
- A protocol for communication and a reference implementation of server software will be created.
- A reference implementation of client software will be developed. In addition to basic functionality, the client software will include a plug-in framework which allows for the integration of a natural language search interface, real-time data extraction of Web content, and semi-automatic lineage linkage.

4 Project Description

This thesis consists of three main aspects: the genealogy data model, the server software and communications protocol, and the client software and initial plug-ins. These aspects will be discussed in the three subsections that follow.

4.1 Data Model

The design of the DFT data model will be strongly influenced by concepts and relationships defined in the *GENTECH Genealogical Data Model*. GENTECH is an effort that represents years of exertion in understanding the genealogical process, and has resulted in a comprehensive logical data model [Lex00]. It is natural and desirable that this logical data model should play a role in the definition of the physical data model for the DFT. However, GENTECH assumes the relational data model, which significantly complicates the representation of data. We will adapt it to work with a graph-centric data model instead, to reduce complexity as well as increase flexibility.

We will use the *Resource Description Framework (RDF)* to physically record genealogical information in the DFT [MM04]. RDF is a graph-centric data model, where a graph consists of a set of statements of the form subject-predicate-object. The subject is always a resource (such as a Web page or a person), and is identified by a URI. The predicate is a resource with specific associated semantics and is also identified by a URI. The object can be either a resource or a literal value (such as a string, an integer, or a date). Statements can be used to record simple facts. For example, the statement “Cassidy is a lawyer” could be represented by the RDF graph shown in Figure 1.

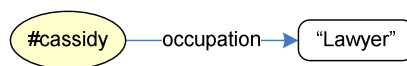


Figure 1: The statement “Cassidy is a lawyer” in RDF

Information recorded in RDF becomes machine-understandable when resources with well-known semantics are used. Words in human languages are often highly ambiguous; the word “love,” for example, has at least eighteen different meanings in the English language.¹ Without a clear and unambiguous definition, it is very difficult for computer software to “understand” what is meant. The semantics of resources can be defined through the use of the *OWL Web Ontology Language*, a vocabulary for specifying the precise meaning of terms and the possible relationships between those terms in RDF. These specifications, called *ontologies*, serve to define a conceptual model for domain-specific information.

We will produce three ontologies that define the fundamental genealogical concepts. The first of these ontologies is the *Genealogy Core (GC)*, which defines how to represent fundamental genealogical facts. The GC models information using two main types of resources: individuals and events.

¹ According to the *American Heritage Dictionary of the English Language, Fourth Edition* (Houghton Mifflin Company, 2004).

- An individual resource is the representation for an actual human being who currently lives or previously has lived. There are several basic predicates defined in the GC for describing an individual, such as name, sex, and occupation.
- An event resource is the representation for a real-world event, such as a birth, marriage, or death. Events typically have an associated date and location.

Relationships between individuals are implicitly represented by associations between individual and event resources. For example, a maternal relationship is recorded by creating a birth event which is related to the child through the `born` predicate and to the mother by the `gaveBirth` predicate. This nuance in the data representation makes possible significantly greater freedom of expression than is typically allowed in more restrictive data formats. For example, it is possible to use an adoption event to associate adoptive parents to a child while simultaneously using a birth event to associate the original parents to the child.

Figure 2 shows an example of modeling genealogical data in this way. The figure represents three fictional individuals: Mark, Sarah, and Samuel Baker. Mark and Sarah are linked together by their marriage in Boston on December 22, 1868. Samuel is linked to Mark and Sarah through his birth event, which occurred five years later in Chicago on April 17, 1873. Light-shaded ovals represent individual resources (with their corresponding URI), while dark-shaded ovals represent event resources. The URIs used in this example, like `#sarah`, `#marriage`, and `#birthOfSamuel`, are for illustrative purposes only. In reality, any arbitrary URI would work just as well, such as `http://www.example.com/genealogy#mark`, or even `uri:uuid:e2f03e8d-7274-4f7d-a9f5-1cab5f9cae80`.

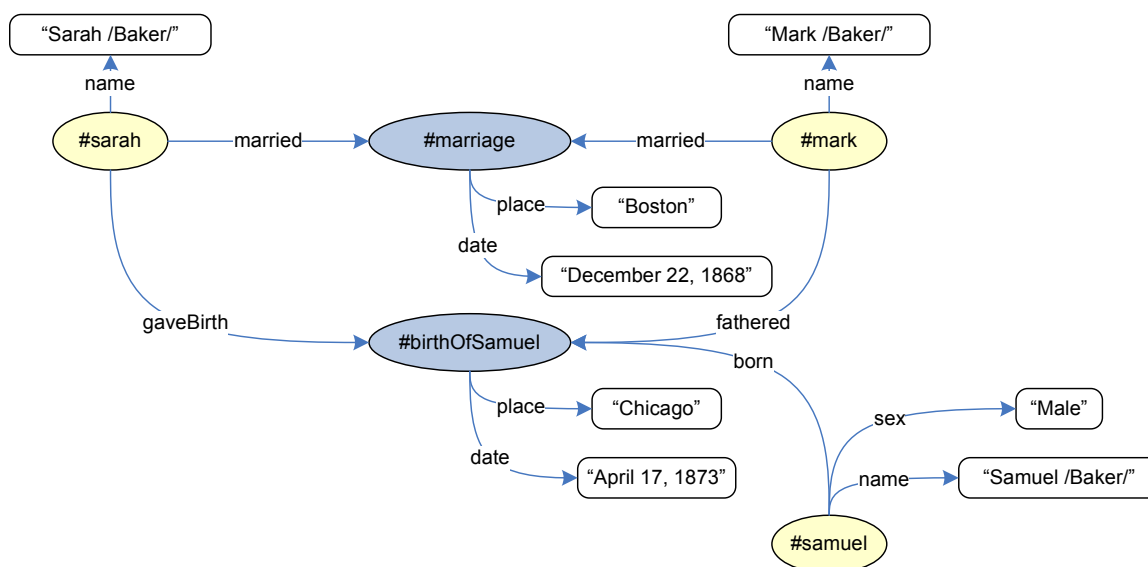


Figure 2: An example of genealogical data in RDF

Any single human being may have more than one individual resource representation. It therefore makes sense to think of an individual resource as a persona, where any number of personas can represent the same human being. The equivalence of personas is recorded by associating them with the `owl:sameAs` predicate. This will prove very useful when data on the same person from

two or more different sources is brought together. It will be the adhesive that brings otherwise isolated nodes of the DFT together into one whole.

It is important to be able to model not just genealogical information, but also its origin. To do so requires the ability to make statements about statements.² *Named graphs* extend the syntax and semantics of RDF to support statements about graphs. A named graph is a closed set of statements identified by a URI. Like any RDF identifier, this URI can be the subject or object of other statements. The second DFT ontology that we will create, called *Genealogy Provenance (GP)*, provides a vocabulary for describing the source of information in a named graph. This vocabulary can be used to annotate not only primary, but secondary sources as well.

In fact, sources can be strung together to form provenance chains. Figure 3 gives an example of one such provenance chain. The first graph, named `#fromBirthCertificate`, contains the name and gender of the fictional William Roberts. A second graph, called `#fromGedcom`, records the source of this data, a birth certificate issued by the city of Detroit. Finally, the source of this second graph indicates that the information in the other two graphs came from a GEDCOM which was imported into the DFT on the fifth of October, 2006. Light-shaded ovals represent individual resources, while dark-shaded ovals represent source resources.

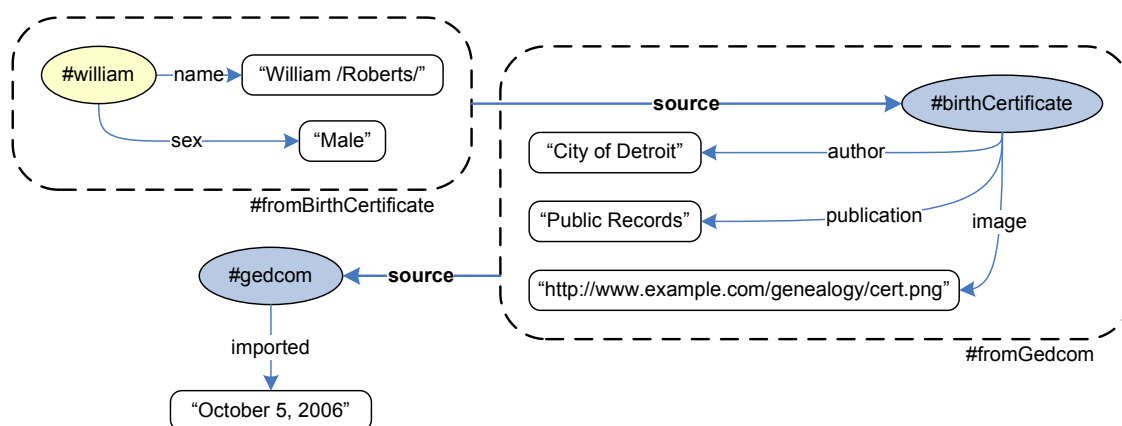


Figure 3: An example of a provenance chain

The open nature of the DFT makes it possible for someone to publish data that claims to come from a different source than it actually does. The *Semantic Web Publishing (SWP)* vocabulary will be used to secure provenance chains [BCM+05]. SWP allows information providers to digitally sign their graphs so that others can be confident about the ultimate source of the information. In order for a genealogist to sign their data, they will need a valid X.509 certificate. Other genealogists will be able to verify the source of their data by confirming that it is signed with the correct certificate. Software takes care of most of this process automatically.

² The RDF recommendation provides for this through a mechanism called reification. However, the semantics of reification are ill-defined, making it an unreliable means of recording the origin of specific information.

Genealogists can specify their degree of trust in information on the DFT at a number of different granularities: statements, graphs, sources, and publishers. Like genealogical information itself, these trust decisions can be published to help inform other genealogists. We will define the *Genealogy Trust (GT)* ontology to support the application and publication of these decisions.

Figure 4 demonstrates the use of this ontology to indicate which information can be trusted and which cannot. This depicts two graphs with information about the fictional John Morris, as well as a third that contains decisions of trustworthiness about the first two. Decisions range from completely untrustworthy (0.0) to totally trustworthy (1.0). Users can define a threshold to decide what information is visible to them. Notice that the upper two graphs give conflicting information on the day of John’s birth. If the user has set a threshold greater than 0.2 then, although the data in the left graph still exists, it does not form part of the owner’s view because its trustworthiness is 0.2.

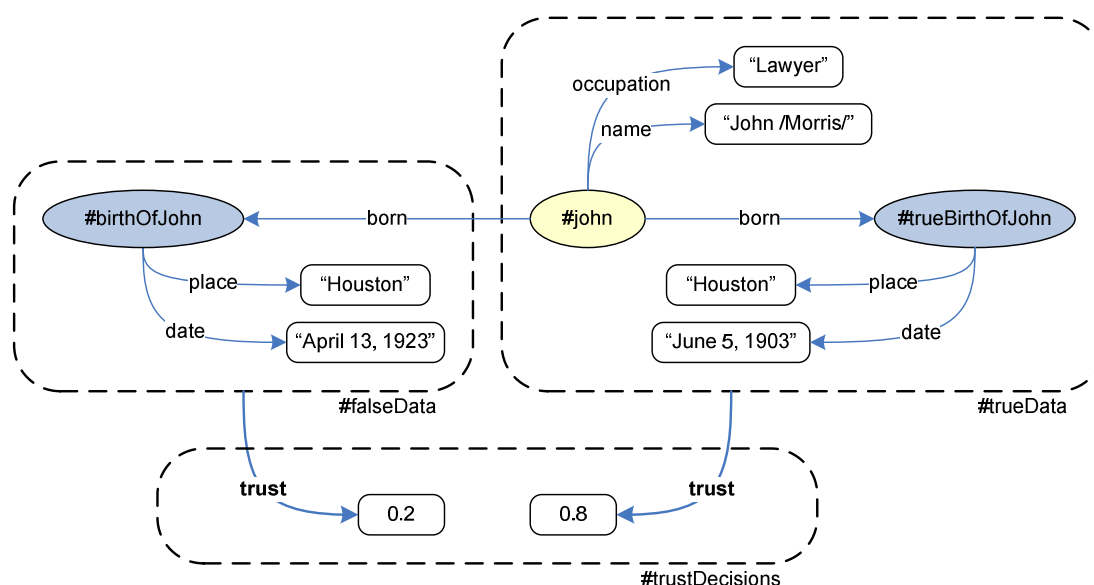


Figure 4: An example of using the *Genealogy Trust* ontology

4.2 Server Software and Protocol

A node of the DFT is simply a Web server that publishes genealogical information according to the established data model and protocol. This means that the proprietors of current resources on the Internet can typically retrofit their servers to support the minimal set of required operations with very few modifications. Nevertheless, servers built from the ground up to serve as nodes are in a good position to offer enhanced capabilities and performance.

Initially we will implement solely a client/server architecture for the DFT. Client software will have access to a directory of DFT nodes, which it can use to search for information one node at a time. Though this approach is simple and effective, we recognize that it fails to scale well. Future work on the DFT should overlay a peer-to-peer architecture so that search queries and responses can scatter and gather across the network in parallel [ADJ+01].

At a high level, the following are the basic operations for the client/server protocol that we will develop.

- *Query*: Client software can search for and retrieve genealogical records with a query request. We will use the *SPARQL Query Language*, a standardized query language for retrieving data from RDF repositories to communicate search requests [PS06].
- *Synchronize*: If the user of the client software has an account with a specific server, the software can synchronize the local data store with the server data store. Graphs of genealogical data are exchanged in either TriG or TriX format [BCW05]. This is the mechanism used to publish data to the DFT.
- *Pingback*: Whenever data that references another server is published, a pingback message can be sent to the other server. This message tells the other server what relevant new data has been added, so that what would otherwise have been a unidirectional connection can become bidirectional.

We will develop server software, code-named *Valhalla*, which implements this protocol. *Valhalla* will be a simple RDF data store that supports partitioned user accounts. It will automatically restrict public access to sensitive information, such as data on living people (defined as anyone born less than 100 years ago without a death on record). Users will be able to have it restrict public access to other information as well.

4.3 Client Software

We will implement a software client, code-named *Genesis*, which consists of an RDF data store and a plug-in framework. Along with *Genesis* we will produce plug-ins that provide manual and automated support for three main activities: data entry, search, and inference. We describe these three activities in detail in the following subsections.

4.3.1 Data Entry

On the simplest level, *Genesis* will function as a record manager that allows users to browse and update genealogical data. This data is maintained in a local data store for efficiency; however, any changes to that data will be synchronized with a designated DFT server.

We will implement the minimal browsing functionality that users might expect from a typical record manager, such as pedigree, index, and individual views. We will also provide the means to manually enter genealogical data, as well as metadata such as the trustworthiness of specific information and sources. The software will help inform trust decisions by making other genealogists' evaluations available in both statistical and detailed form.

Because manual data entry is slow, tedious, and error prone, we will also provide automated means for data entry. To do this, *Genesis* will include a simple, integrated Web browser which users can use to find genealogical information through conventional means on the Web. This browser will include an "import" button which, when clicked, will automatically extract the data from the current Web page, display that data to the user for verification and possible correction, and then import that data to the local data store.

The data is extracted with a technique developed by the Data Extraction Group at BYU that probes the Web page with respect to an *extraction ontology* [Wal04]. An extraction ontology is

a conceptual model for domain-specific information, coupled with regular expressions which help locate and identify pieces of information and relationships in that domain. These pieces, once identified, can be lifted from the surrounding text and markup and reassembled into RDF.

4.3.2 Search

The rich structure of genealogical data in the DFT is one of its greatest advantages, because it allows researchers to perform very specific, complex queries to find exactly what they seek. Users will not benefit from this if they have to expend extra effort to prepare formal queries, however. With this in mind, we will write a plug-in for *Genesis* that accepts natural language queries, such as “Find anyone with the last name Marcus, born near Boston in 1893,” and converts them to formal queries. Furthermore, these queries do not need to be complete sentences, let alone grammatically correct. For example, the user could enter “Rodger Davis, plumber, died 1972,” or even the perfunctory “Grandma.”

The plug-in will translate natural language queries into SPARQL with a modified³ version of the *AskOntos* system [Vic06]. *AskOntos* transforms a query by analyzing it with respect to an extraction ontology. The system identifies key words and phrases from the original query and then combines them to produce a formal query that can be applied to structured data in the domain. Once the user’s query has been translated into SPARQL, the plug-in submits the query to each of the DFT server nodes in the directory, one-by-one. As results come back, they are stored in the local cache and displayed to the user.

We will also include a user agent plug-in that runs at startup and gathers additional genealogical information in the background. This agent will iterate over all the individuals in the local data store, automatically generate search queries, and submit them to nodes on the DFT for more information. These search queries will be a simple combination of a few of an individual’s known attributes. For example, if all that is known about a given individual is her name and birth date, the agent will search for individuals with that same name and/or birth date. As results come back, they will be displayed to the user and stored in the local cache. A second user agent, which we will describe in the next section, will perform further analysis on these results.

4.3.3 Inference

Most genealogy software provides a “merge” function, which allows the user to compare and combine similar individuals. This is usually a destructive modification where data is overwritten and lost. The DFT supports a similar but non-destructive notion, the idea of declaring two or more personas “equivalent” through the use of the `owl:sameAs` predicate. In this way the two individuals are logically merged, while their physical records remain separate.⁴

To support this mechanism, *Genesis* will allow the user to manually select two individuals for comparison. The software will display the information on these two individuals side-by-side and

³ The original *AskOntos* system translates natural language queries into the *XQuery* language, which is unsuitable for searching particularly large RDF data stores.

⁴ This does not preclude the possibility of manual destructive merges, which can improve performance for those cases where two or more personas are judged by the owners of the data to be unmistakably identical.

allow the user to decide whether they are the same. If the user believes that they are indeed the same, the user can click a button that will logically merge the two.

To assist users with this decision, we will implement the lineage linkage technique developed by the Data Mining Lab at BYU [Pix06]. This technique accepts as input the attributes of the two individuals, as well as the attributes of their close relations. It then runs the attributes through a neural network that has been specifically trained for the purpose of comparing individuals, which outputs a measure of their degree of similarity. Finally, it uses a user-defined threshold to decide whether the two individuals represent the same person. While this technique is reasonably accurate, it is not perfect. Rather than fully automate the procedure, the software will only make a recommendation to the user, who can then analyze the evidence and intelligently assess whether the inference is correct, incorrect, or undecidable at this time.

As the utility of the DFT will increase with greater connectivity, we would like to facilitate the establishment of relationships between isolated pedigrees. To do this, we will include a user agent plug-in that runs at startup and performs semi-automatic lineage linkage in the background. It will iterate over the individuals in the local data store and use a simple heuristic to compare each of them for similarity with each of the other individuals in both the local data store and cache. Whenever a pair of relatively similar individuals is found, the two will undergo the more rigorous neural network comparison. The plug-in will notify the user if any pair is found to be significantly similar, at which time the user can elect to compare the two and logically merge them.

4.4 Technology

We will write the software for this thesis in Java to take advantage of the open-source support for Semantic Web technologies (RDF, OWL, SPARQL, etc.) available on that platform. In addition to the actual source code used in the work of BYU researchers previously cited, we will also use the following open source components.

- The *Jena Semantic Web Framework*, which supports RDF, OWL, and inference reasoning [Jen06].
- *Named Graphs API for Jena (NG4J)*, which extends Jena with named graph and SWP support [BCW05].
- *Apache Lucene*, for advanced text searching capabilities [Hat05]. In particular, this will enable indexed search of names and other facts not only by exact match, but by wildcard expressions, phonetic algorithms (such as Soundex), and Levenshtein distance as well.
- *MySQL*, for the storage of RDF graphs and statements [MyS06]. MySQL is actually a relational database system, but it can be used for graph-based data with the translation layers provided by Jena and NG4J.
- *Eclipse Rich Client Platform*, which provides a Java framework for the plug-in application model [ML05]. It also provides a GUI framework that enables the native look-and-feel of the host operating system.
- *JUnit* will be used as a framework for writing functional, unit, and integration tests [JUn06].

5 Validation

Our thesis calls for the specification of a data model, communications protocol, server software, and client software that supports a plug-in framework. We will validate these deliverables with a test installation of *Valhalla* on five different servers. We will connect to each server in turn with *Genesis* to store a different subset of a body of genealogical, provenance, and trust information. We will then establish links between the servers and show that the distributed data can be seamlessly browsed. Finally, we will show that the included plug-ins function to demonstrate the plug-in framework in *Genesis*.

In order to assure a high quality of work, we will follow modern software development practices throughout the course of this thesis.

- We will write a high-level design document consistent with the specification in this thesis proposal.
- We will keep all source code and documentation under version control and regularly backed up.
- The source code itself will conform to standard Java code conventions and be appropriately commented.
- We will write functional, unit, and integration tests to a practical level of code coverage and use them for frequent regression testing.
- We will provide external documentation in the form of a quick-start guide, an online help system, a plug-in extension writer's guide, and JavaDocs.

All of this documentation and source code will serve as formal project deliverables.

6 Thesis Schedule

Following is the proposed schedule for completion of this thesis.

Milestone	Deadline
High-Level Design Document	November 2006
Implementation:	
Client Software: Data Entry	December 2006
Client Software: Search	March 2007
Client Software: Automated Analysis	May 2007
Server Software	July 2007
Documentation	August 2007
Submit Thesis to Advisor	October 2007
Submit Thesis to Committee Members	October 2007
Thesis Defense	November 2007

7 Annotated Bibliography

- [ADJ+01] C. C. Albrecht, D. Dean, R. B. Jackson, S. W. Liddle, and R. D. Meservy. A Peer-To-Peer Network Protocol for Genealogical Data. In *Proceedings of the First Family History Technology Workshop*, pages 19–23, Provo, Utah, April 2001.

Gives a high-level description of the *Genealogy Network Transfer Protocol (GNTP)*, a peer-to-peer network protocol which was proposed, but not completed, for publishing and searching genealogical information. The protocol developed for the DFT in this work will be strictly server/client; however, a peer-to-peer network protocol such as GNTP should be added in some future work.

- [BCM+05] C. Bizer, R. Cyganiak, O. Maresch, and T. Gauss. TriQLP—Trust Architecture. Free University of Berlin, January 2005. (sites.wiwiss.fu-berlin.de/suhl/bizer/TriQLP).

Provides the schema and examples of the *Semantic Web Publishing (SWP)* vocabulary, used to publish digitally signed named graphs. The SWP will be used to secure provenance chains.

- [BCW05] C. Bizer, R. Cyganiak, and R. Watkins. NG4J—Named Graphs API for Jena. Free University of Berlin, October 2005. (sites.wiwiss.fu-berlin.de/suhl/bizer/ng4j).
- Introduction, examples, downloads, and online documentation for the *Named Graphs API for Jena (NG4J)*, an “extension to the Jena Semantic Web framework for parsing, manipulating, and serializing sets of Named Graphs.” This library will be used to implement named graphs in *Valhalla* and *Genesis*.
- [CBH+05] J. J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named Graphs, Provenance and Trust. In *Proceedings of the 14th International Conference on World Wide Web*, pages 613–622, Chiba, Japan, May 2005.
- An extension of the syntax and semantics of RDF to support RDF graphs nameable by URI, which are used in the DFT to model provenance.
- [Hat05] E. Hatcher. *Lucene in Action*. Manning Publications, New York, New York, 2005.
- Documentation and examples for the *Apache Lucene* library, a high-performance, full-text search engine. This library will be used in *Valhalla* and *Genesis* to provide quick access to data in underlying RDF data stores.
- [JL01] E. Jarvi and N. Leippe. Global Genealogy Network Extended Abstract. In *Proceedings of the First Family History Technology Workshop*, pages 15–18, Provo, Utah, April 2001.
- Gives a high-level description of the *Global Genealogy Network*, a distributed network of genealogical data which was proposed but not completed.
- [Jen06] Jena Semantic Web Framework, May 2006. (jena.sourceforge.net).
- Documentation, examples, and downloads for the *Jena Semantic Web Framework*, a Java library which offers support for RDF, OWL, and SPARQL. This library will be used in *Valhalla* and *Genesis* to store, manipulate, and search genealogical data in RDF form.
- [JUn06] JUnit, Testing Resources for Extreme Programming, October 2006. (www.junit.org).
- Documentation, examples, and downloads for the *JUnit*, a Java testing framework which will be used to regression test *Valhalla* and *Genesis*.
- [Lex00] Lexicon Working Group. GENTECH Genealogical Data Model v1.1. Technical report, National Genealogical Society, May 2000. (www.ngsgenealogy.org/ftp/pub/GENTECH_Data_Model/Description_GENTECH_Data_Model_1.1.doc).
- Definition of the *GENTECH Genealogical Data Model*, a comprehensive data model for genealogical research and analysis. The design of the DFT data model will be heavily based on the GENTECH model.

- [MM04] F. Manola and E. Miller, editors. *RDF Primer*. Technical report, World Wide Web Consortium, February 2004. (www.w3.org/TR/rdf-primer).
- Provides the basic knowledge necessary to understand and use the *Resource Description Framework* (RDF), the foundation for data storage in the DFT.
- [ML05] J. McAffer and J.-M. Lemieux. *Eclipse Rich Client Platform: Designing, Coding, and Packaging Java Applications*. Addison-Wesley Professional, Boston, Massachusetts, 2005.
- Guide to writing and deploying Java applications using the *Eclipse Rich Client Platform* (RCP). This platform will be used in *Genesis* for the supplied plug-in framework.
- [MvH04] D. L. McGuinness and F. van Harmelen, editors. *OWL Web Ontology Language Overview*. Technical report, World Wide Web Consortium, February 2004. (www.w3.org/TR/owl-features).
- An informal introduction to the features and capabilities of the *OWL Web Ontology Language*, used in the DFT to define the schema for genealogical data.
- [MyS06] MySQL AB :: The world's most popular open source database, October 2006. (www.mysql.com).
- Documentation and downloads for MySQL, a fast, reliable, and easy-to-use open source database system. This database system will be used for the physical storage of RDF data in *Valhalla* and *Genesis*.
- [Pix06] B. Pixton. *Improving Record Linkage Through Pedigrees*. Master's Thesis, Brigham Young University, Provo, Utah, July 2006.
- Thesis on the use of a filtered, structured neural network which exploits the family relationships between individuals to identify individuals shared between pedigrees. This technique will be used in a plug-in for *Genesis* to perform semi-automatic lineage linkage.
- [PS06] E. Prud'hommeaux and A. Seaborne, editors. *SPARQL Query Language for RDF*. Technical report, World Wide Web Consortium, April 2006. (www.w3.org/TR/rdf-sparql-query).
- Describes the SPARQL query language for RDF, which will be used in the DFT communication protocol to search RDF data stores.
- [Qua03] D. Quass. *Keynote Address: Perspectives on Research Problems in Family History from the LDS Family and Church History Department*. *Third Family History Technology Workshop*, Provo, Utah, April 2003.
- Discusses their concentrated effort to make family history easier for the "non-genealogist." The idea of a common pedigree and world record manager is highlighted.

- [See00] D. A. Seeley. Network evolution and the emergence of structure. In T. R. J. Bossomaier and D. G. Green, editors, *Complex Systems*, pages 51–89, Cambridge, 2000. Cambridge University Press.

Describes the *double jump* phenomenon, wherein the overall connectivity of a graph dramatically increases when a critical number of random connections have been added. The DFT relies on this phenomenon to achieve near-complete connectivity of the human family tree with minimal effort.

- [Vic06] M. Vickers. Ontology-Based Free-Form Query Processing for the Semantic Web. Master's Thesis, Brigham Young University, Provo, Utah, June 2006.

Introduces the *AskOntos* system, which uses information extraction to transform a natural language query into a formal, machine-understandable query. This technique will be used in a plug-in for *Genesis* to provide a natural language query interface.

- [Wal04] T. Walker. Automating the Extraction of domain-Specific Information from the Web – A Case Study for the Genealogical Domain. Master's Thesis, Brigham Young University, Provo, Utah, November 2004.

Describes *GeneTIQS*, a system which automatically extracts data from the Web that is scalable and resilient to changes in the content and structure of the underlying documents. This technique will be used in a plug-in for *Genesis* to extract genealogical information from the Web.

- [Woo01] S. N. Woodfield. Towards Effective, Real-Time Collaboration In Genealogy Research. In *Proceedings of the First Family History Technology Workshop*, pages 28–31, Provo, Utah, April 2001.

Proposes a peer-to-peer virtual database to enable real-time collaboration and private views that allow users to see the database from their preferred perspective.

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis proposal submitted by

Hilton Campbell

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

_____	_____
Date	David Embley, Chair
_____	_____
Date	Cristophe Giraud-Carrier
_____	_____
Date	Tony Martinez
_____	_____
Date	Parris K. Egbert
	Graduate Coordinator