# Automating the Extraction of Genealogical Information from the Web

*Troy Walker*
*David W. Embley*
*Department of Computer Science*
*Brigham Young University*
*{troywalk, embley}@cs.byu.edu*

## Introduction

Thousands of amateur genealogists have posted the results of their research online. Hundreds of organizations including governments, companies, and churches make their vital records databases available for querying. All together, there are probably over a million pages containing genealogical data on the Web in addition to countless pages generated from database queries.

Search engines can help genealogists locate information of interest, but since search engines do retrieval on a whole-document basis, a genealogist must manually load each returned page and either read it for the precise data wanted, or determine that the document is irrelevant. For example, a search on Google for "Walker genealogy" returns over 200,000 results. Although a few of the pages are advertisements for genealogical services and are therefore irrelevant to the query, most of these results are relevant and represent a wealth of information. Unfortunately, the amount of time necessary to visit all these sites is discouraging: Assuming a human could read each of these pages in one minute and work 24 hours a day, 7 days a week, it would take over four months to read all this information.

To help computer users sift through this mountain of data, we have built a tool to extract genealogical data from arbitrary Web pages and store it in a database format. This is accomplished using (1) an extraction ontology describing genealogical data, (2) a record separator using vector space modeling techniques to split Web pages into genealogical records, and (3) a result presenter. In the remainder of this extended abstract, we explain each of these three components of the system and then present some evaluation data and make concluding remarks.

## Extraction Ontology

The Data Extraction Group (DEG) at Brigham Young University has developed a platform for using ontologies (or conceptual models) to extract data from restricted domains [ECJ+99]. In the DEG approach, we specify the elementary data items to extract by data frames [Emb80] that

{CapPhrase}\,( +){CapPhrase}\,( +){State}
{CapPhrase}\,( +){State}
{CapPhrase}\,( +){CapPhrase}\,( +){CapPhrase}\,( +){Country}
{CapPhrase},( +){CapPhrase},( +){Country}
{State}
CapPhrase: (([A-Z][A-Za-z]*)|of|the|on)(\s+(([A-Z][A-Za-z]*)|of|the|on)){0,3}

**Figure 1: Regular Expression for Locations**

include regular expressions and possible keywords. Figure 1 shows regular expressions for locations. They use lexicons for states and countries, which are lists that include all known spellings and abbreviations of all states and countries. Further, the regular expressions use the macro *CapPhrase*, which recognizes phrases of capitalized words and other words common in place names. Ontologies also express relationships between data extracted, using participation constraints to specify the minimum, average, and maximum number of relationships in which an

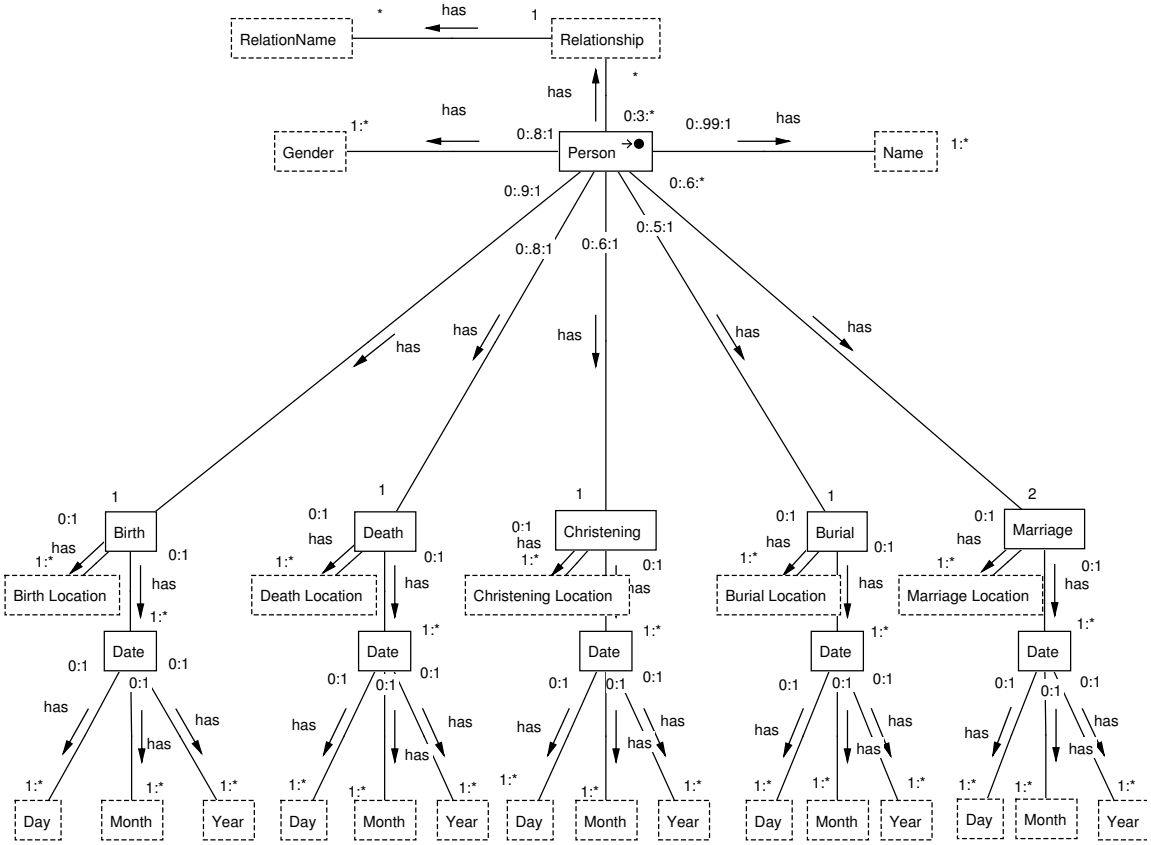


**Figure 2: Genealogy Ontology**

item of data can participate. Figure 2 shows our ontology for genealogy. Each person has a name and gender and a part in many events such as birth, marriage, and death. Each event has date and location attributes. The participation constraints for person in the relationship between person and birth declare that in a genealogical record, we may or may not find birth information (minimum=0 and maximum=1) but that we expect to find birth information about 90% of the time (average=.9).

The DEG extraction engine extracts data that conforms to a specified ontology. For a given single record, the data conforms if it satisfies the data-frame recognizers, and the relationships among the data items conform if they satisfy the participation constraints. This approach has the advantage of being resilient to changes in page format and is capable of extracting from new pages with no human intervention. This makes it ideal for the genealogy application domain, where pages come from a variety of authors, pages may be updated often, and new pages are continually brought online.

## Record Separator

In order to obtain the records from which we can extract data that conforms to an ontology, we need to know when the information about one person ends and another begins. Record separation techniques proven to work well in domains such as car ads, job listings, and obituaries (see [EJN99]) do not work well with genealogy pages. Genealogical Web sites created by hobbyists are extremely diverse and often employ complicated markup to show family tree structures. Figure 3a shows an example of a document that contains information on one person. The example in Figure 3b has many records that follow a regular pattern, which is easy to discover. The records in 3c, although easily separated by a human genealogist, are difficult to separate by a computer program not built specifically "understand" family trees. Accurately separating records is the major challenge in extracting genealogical data. Once we have the data for only one person, it is relatively easy to do the extraction (provided we have good recognizers). We build our record separator to handle all of these cases and any others we might encounter such as links to sub-records and factored data.

Along the lines discovered in [LX00], we have expanded [EJN99] to make use of vector space modeling (VSM) to separate records. From the ontology, we build a vector representing a prototypical record where each dimension of the vector is the average number of occurrences of an attribute as found in the ontology's participation constraints. We then build a DOM tree [WHA+00] representing each document from the HTML markup. Each node *N* of this DOM tree has a vector recording the data frame matches found within the subtree rooted at *N*. We calculate

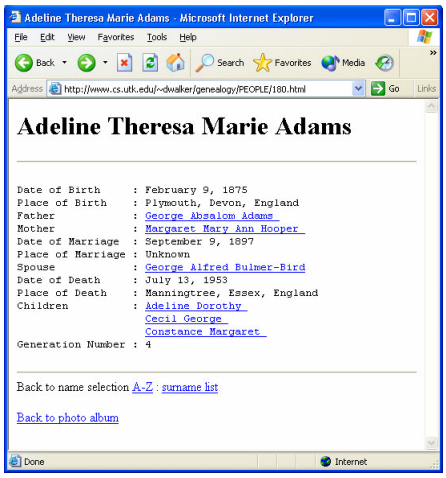


**Figure 3a: Single Record Document**

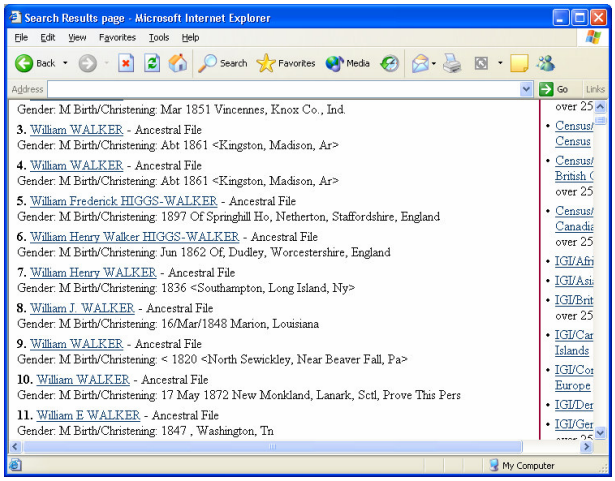


**Figure 3b: Simple Multiple-Record Document**

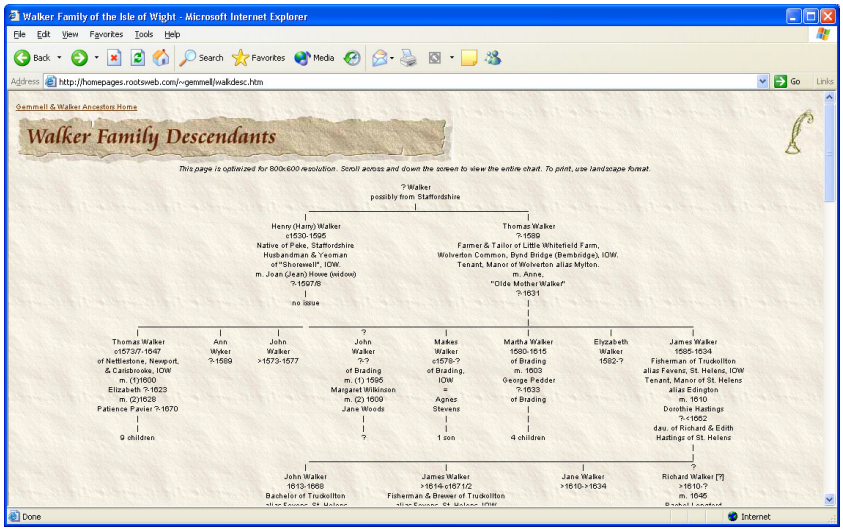


**Figure 3c: Complex Multiple-Record Document**

the similarity between a node of the DOM tree and our prototypical record by computing the cosine of the angle between the respective vectors. Our record separator starts at the root of the tree and works toward the leaves, exploring branches that have high cosine measures and either pruning those with low scores or folding them into other subtrees. We split subtrees with high vector magnitudes using heuristics from [EJN99].

## Result Presenter

Once a document is split into records and extracted with respect to the ontology, the data is ready to be displayed. Our system builds HTML tables from the extracted data that is then shown in the

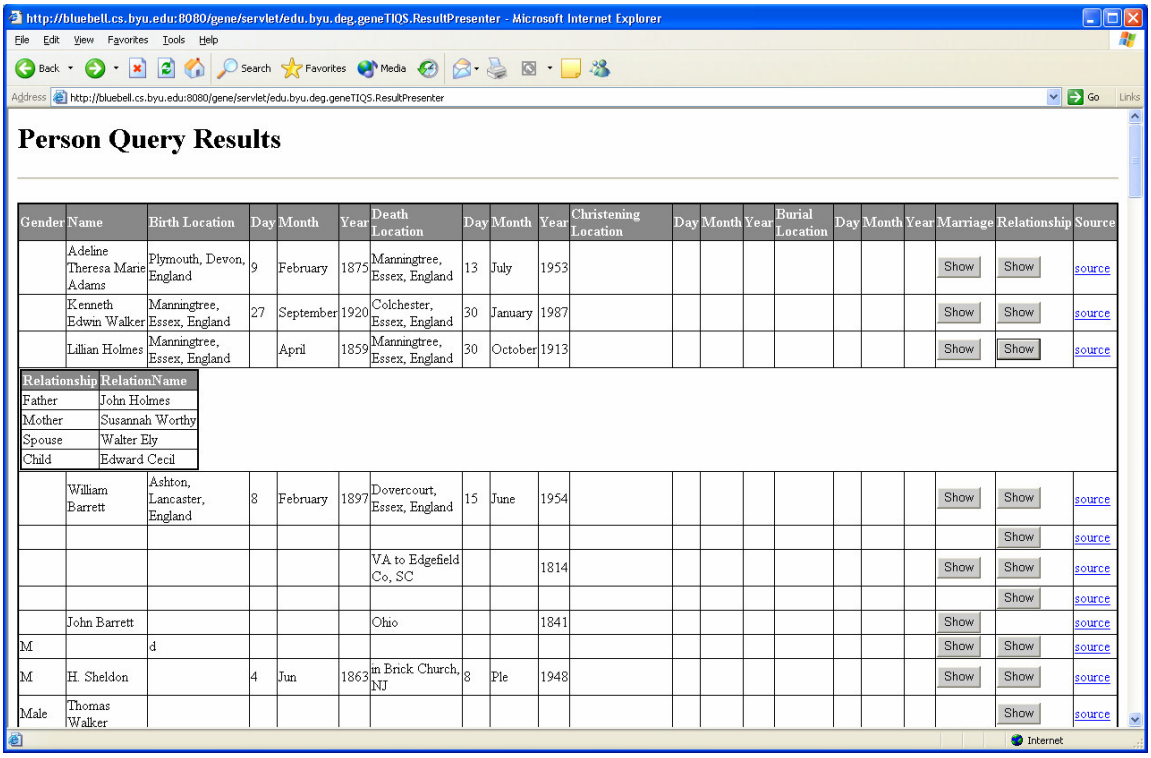


**Figure 4: Results Presented to User**

user's Web browser (see Figure 4 below). Attributes that may have multiple values, such as relationships and marriages, are hidden in subtables and viewed as needed by selecting the *Show* button in the row and column. Figure 4, for example, shows the relationships of Lillian Holmes, which appeared when we clicked on a *Show* button in Lillian's row and the *Relationship* column.

## Results

To evaluate the record separation part of our system, we tested it on documents representing three classes of genealogical Web pages following the example pages in Figure 3: single record, simple record, and complex record. For each document in our test set, we determined, by hand, the correct separation of records. We then ran our record separator and compared the results. A

| | Single Record | Simple Record | Complex Record |
|---|---|---|---|
| Number of pages | 10 | 3 | 3 |
| Number of records | 10 | 268 | 159 |
| Precision | 94.1% | 97.3% | 93.6% |
| Recall | 100% | 94.7% | 88.3% |

**Table 1: Experimental Results**

resulting record was considered correct if it contained all of the information for that person found on the original document. Within each class of document, we obtained precision and recall measures as shown in Table 1. The richness of data found in the single record documents tend to make the record separator split single records while the scarcity of data in the other categories tend to make it discard legitimate records. As we improve on these initial results, we will refine this balance.

## Conclusion

We have developed a tool to separate the records in genealogical Web documents and extract the data found within them. Our initial results are promising, and we will continue to develop and refine our technique.

## References


[ECJ+99] D.W. Embley, E.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, and R.D. Smith. Conceptual-Model-Based Data Extraction from Multiple-Record Web Documents. *Data and Knowledge Engineering,* 31(3): 227-251, November 1999.

[EJN99] D.W. Embley, Y.S. Jiang, and W.-K. Ng. Record-Boundary Discovery in Web Documents. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 467-478, Philadelphia, Pennsylvania, June 1999.

[Emb80] D.W. Embley. Programming With Data Frames for Everyday Data Items. In *Proceedings of the 1980 National Computer Conference*, pages 301–305, Anaheim, California, May 1980.

[WHA+00] L. Wood, A. Le Hors, V. Apparao, S. Byrne, M. Champion, S. Isaacs, I. Jacobs, G. Nicol, J. Robie, R. Sutor, and C. Wilson (Eds.). *Document Object Model (DOM) Level 1 Specification*, http://www.w3.org/TR/2000/WD-DOM-Level-1-20000929/, September 2000.

[EX00] D. W. Embley, and Li Xu. Locating and Reconfiguring Records in Unstructured Multiple-Record Web Documents, In *Proceedings of the Fifth International Workshop on the Web and Databases(WebDB 2000),* pages 123-128, Dallas, Texas, May 2000.