FROntIER: A Framework for Extracting and Organizing Biographical Facts in Historical Documents

Joseph Park

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

David W. Embley, Chair
Stephen W. Liddle
Charles D. Knutson

Department of Computer Science

Brigham Young University

January 2015

ABSTRACT

FROntIER: A Framework for Extracting and Organizing Biographical Facts in Historical Documents

Joseph Park
Department of Computer Science, BYU
Master of Science

The tasks of entity recognition through ontological commitment, fact extraction and organization with respect to a target schema, and entity deduplication have all been examined in recent years, and systems exist that can perform each individual task. A framework combining all these tasks, however, is still needed to accomplish the goal of automatically extracting and organizing biographical facts about persons found in historical documents into disambiguated entity records. We introduce FROntIER (**F**act **R**ecognizer for **Ont**ologies with **I**nference and **E**ntity **R**esolution) as the framework to recognize and extract facts using an ontology and organize facts of interest through inferring implicit facts using inference rules, a target ontology, and entity resolution. We give two case studies of FROntIER's performance over a few select pages from *The Ely Ancestry* [BEV02] and *Index to The Register of Marriages and Baptisms in the Parish of Kilbarchan, 1649–1772* [Gra12].

Keywords: information extraction, inference, entity resolution

ACKNOWLEDGMENTS

Thank you to all those who have helped accomplish the great task of completing this thesis. I express my appreciation to the readers of this thesis for taking the time to look at it. I express even more gratitude to the following individuals and groups:

To the late Ki Sung Park, my father, who has and does still inspire me in all aspects of my life. We started this journey together and the memory of him has kept me motivated through the difficult times. He is missed greatly.

To my advisor Dr. David W. Embley, who took me under his wing while I was a young undergraduate and has mentored me through the entire process of this thesis (especially after his retirement since he volunteered his time to help me). I would also like to thank him for helping me to finish this thesis and not quit.

To the members of the Data Extraction Research Group, current and former, whose works have laid the foundation for this thesis and whose tools aided in completing it.

To my friends and family members, who supported my efforts and encouraged me to stay on task.

Table of Contents

List of Figures

Chapter 1

Introduction

Historians, genealogists, and others have great interest in gaining knowledge about people and places from historical documents through fact extraction and organization. Figure 1.1, for example, shows a page from *The Ely Ancestry* [BEV02] and is representative of the type of knowledge and documents desired. Facts of interest in the figure include those explicitly stated such as *William Gerard Lathrop was born in 1812*[1], *married Charlotte Brackett Jennings in 1837*, and *is the son of Mary Ely*. In addition to explicitly stated facts, implicit facts are also of interest. These include the fact that *William Gerard Lathrop is male*, inferred from the stated fact that he is a son, and *Maria Jennings has surname Lathrop*, inferred from cultural tradition and the stated fact that her father has the surname Lathrop. Implicit facts also include disambiguating references to objects. An example of reference disambiguation in Figure 1.1 is that the first Mary Ely mentioned on the page and the third Mary Ely mentioned are the same person, but not the same person as the second-mentioned Mary Ely, since the first-mentioned Mary Ely is the mother of Abigail while the second-mentioned Mary Ely is Abigail's daughter.

Automating the process of extracting stated facts, inferring implicit facts, and resolving object references is a difficult task. Sarawagi [Sar08] surveys much of the work of the last decade or so that has been done in automated information extraction of facts from unstructured and semi-structured text. For inferring implicit facts, work dates back to Aristotle and is typified nowadays by the work in description logics [BCM+03], which describes research

[1]Explicit facts have been syntactically rearranged and unabbreviated from their original format in the document to make them readable. Implicit facts have likewise been modified from their original format.

SEVENTH GENERATION.

241213. Mary Eliza Warner, b. 1826, dau. of Samuel Selden Warner and Azubah Tully; m. 1850, Joel M. Gloyd (who was connected with Chief Justice Waite's family).

243311. Abigail Huntington Lathrop (widow), Boonton, N. J., b. 1810, dau. of Mary Ely and Gerard Lathrop; m. 1835, Donald McKenzie, West Indies, who was b. 1812, d. 1839.

(The widow is unable to give the names of her husband's parents.)

Their children:

1. Mary Ely, b. 1836, d. 1859.
2. Gerard Lathrop, b. 1838.

243312. William Gerard Lathrop, Boonton, N. J., b. 1812, d. 1882, son of Mary Ely and Gerard Lathrop; m. 1837, Charlotte Brackett Jennings, New York City, who was b. 1818, dau. of Nathan Tilestone Jennings and Maria Miller. Their children:

1. Maria Jennings, b. 1838, d. 1840.
2. William Gerard, b. 1840. } Twins.
3. Donald McKenzie, b. 1840, d. 1843. }
4. Anna Margaretta, b. 1843.
5. Anna Catherine, b. 1845.

243314. Charles Christopher Lathrop, N. Y. City, b. 1817, d. 1865, son of Mary Ely and Gerard Lathrop; m. 1856, Mary Augusta Andruss, 992 Broad St., Newark, N. J., who was b. 1825, dau. of Judge Caleb Halstead Andruss and Emma Sutherland Goble. Mrs. Lathrop died at her home, 992 Broad St., Newark, N. J., Friday morning, Nov. 4, 1898. The funeral services were held at her residence on Monday, Nov. 7, 1898, at half-past two o'clock P. M. Their children:

1. Charles Halstead, b. 1857, d. 1861.
2. William Gerard, b. 1858, d. 1861.
3. Theodore Andruss, b. 1860.
4. Emma Goble, b. 1862.

Miss Emma Goble Lathrop, official historian of the New York Chapter of the Daughters of the American Revolution, is one of the youngest members to hold office, but one whose intelligence and capability qualify her for such distinction. Miss Lathrop is not without experience; in her present home and native city, Newark, N. J., she has filled the positions of secretary and treasurer to the Girls' Friendly Society for nine years, secretary and president of the Woman's Auxiliary of Trinity Church Parish, treasurer of the St. Catherine's Guild of St. Barnabas Hospital, and manager of several of Newark's charitable institutions which her grandparents were instrumental in founding. Miss Lathrop traces her lineage back through many generations of famous progenitors on both sides. Her maternal ancestors were among the early settlers of New Jersey, among them John Ogden, who received patent in 1664 for the purchase of Elizabethtown, and who in 1673 was

Figure 1.1: Page 419 of *The Ely Ancestry.*

on methods for defining first-order logics, proving soundness and decidability, and inferring facts from existing facts. To help disambiguate object references—solve the record linkage or entity resolution problem—researchers often resort to the use of statistical methods, which include machine learning algorithms [Chr12]. Though much has been accomplished and still more can be done to thoroughly examine these issues, what is lacking most is tying them together into a unified, synergistic whole—a framework.

In answer to this lack of a unifying framework, we have created FROntIER (**F**act **R**ecognizer for **Ont**ologies with **I**nference and **E**ntity **R**esolution) as a framework to automatically extract and organize facts about people found in historical documents. FROntIER makes use of extraction ontologies [ECJ+99, ELL11] to automatically extract stated facts of interest using regular expression patterns and dictionaries. Once stated facts of interest have been recognized and properly associated with an extraction ontology, FROntIER disambiguates objects, infers additional facts about these objects, and organizes the objects and facts about these objects with respect to a target ontology.

FROntIER's extraction ontologies allow users to model text and layout as it appears in historical documents, while FROntIER's target ontologies model knowledge of interest to be gleaned by historians—facts both directly and indirectly stated. To see the difference, compare the target ontology in Figure 1.2, which is an ontological view of biographical facts of a person, against the extraction ontology in Figure 1.3, which models how explicitly stated biographical facts appear in *The Ely Ancestry*. FROntIER uses pattern-based extractors (recognizers) to identify the existence of objects and their interrelationships according to the particular layout in the text document, and uses logic rules to organize extracted facts in a target ontology. FROntIER, for example, extracts the stated "son of" and "dau. of" facts into the *Son-Person* and *Daughter-Person* relationship sets in Figure 1.3 and then uses the inference rules "if Son, then male" and "if Daughter, then female" to populate the *Person-Gender* relationship set in Figure 1.2. Inference and organization also include entity resolution, which proceeds based on extracted and inferred facts. The first-mentioned Mary

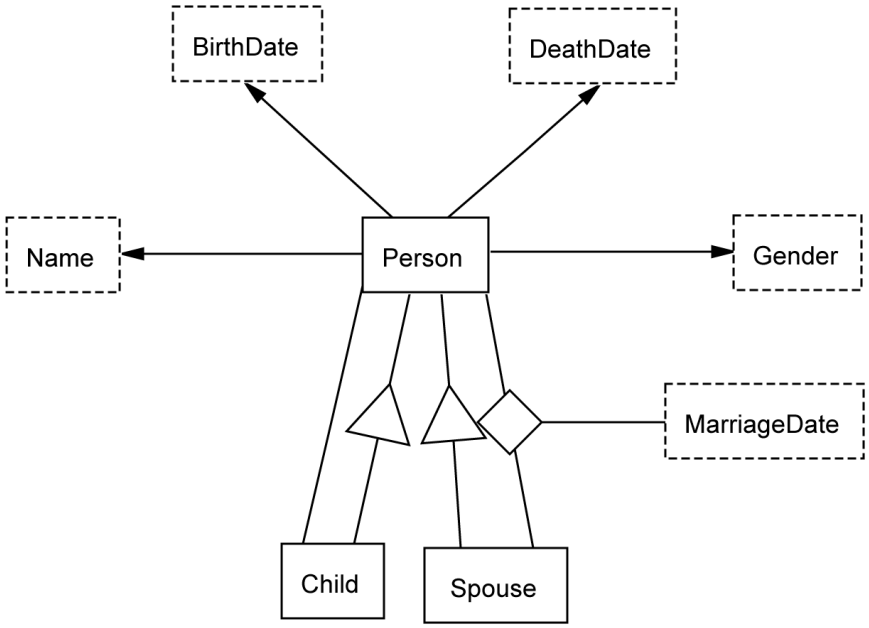

Figure 1.2: Target Ontology of Desired Biographical Facts.

Ely in Figure 1.1, for example, is the grandmother of the second-mentioned Mary Ely, and therefore cannot be the same Mary Ely.

The contribution of this thesis is the construction of a unified framework for extracting and organizing facts that includes:

1. Provisions for users to express relationship-based regular-expression extractors and record-based regular-expression extractors (in addition to the already existing entity-based regular-expression extractors);
2. Provisions for users to state object existence rules for identifying the existence of objects such as people;
3. Provisions for users to specify inference rules for obtaining inferred facts; and
4. Provisions for automatic, fact-based entity resolution.

With these framework provisions, FROntIER is able to extract and organize both stated and implied facts found in OCRed historical documents.

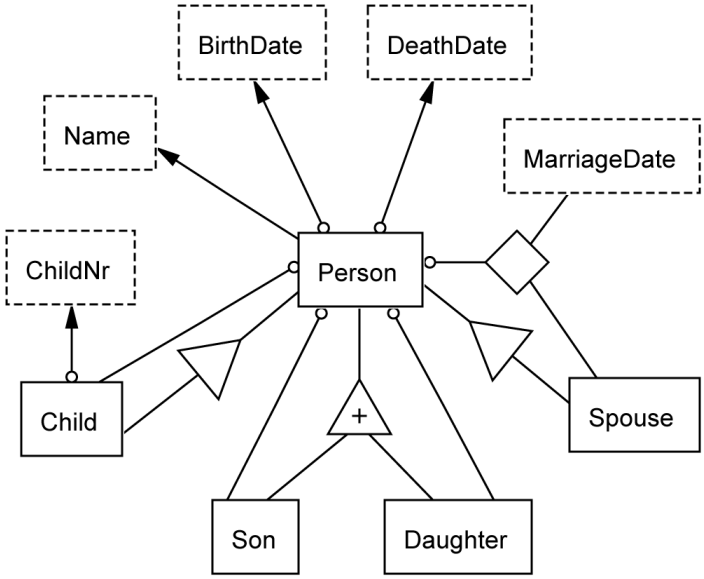

Figure 1.3: Source Extraction Ontology of Stated Biographical Facts in *The Ely Ancestry*.

The rest of the thesis proceeds as follows. Chapter 2 compares FROntIER to related systems and similar work. Chapter 3 explains the basics of extraction ontologies and gives the details about the extractor types designed for this thesis. Chapter 4 details how inference is used in FROntIER to produce implicit facts and organize them from a source ontology to a target ontology. Chapter 5 explains the use of object identity resolution to produce disambiguated entity records. Chapter 6 includes two case studies of using FROntIER to process OCRed pages from historical documents. Chapter 7 concludes the thesis with a summary of the contributions and future work.

Chapter 2

Related Work

FROntIER spans three areas of research: information extraction, logical reasoning, and entity disambiguation. We know of no work that spans all three areas in a unified framework for accomplishing the task of extracting and organizing information from text documents. A few research efforts focus on both automatic fact extraction and record linkage (e.g. [GX09], [BGH09], [BBC+10], [BHH+11]). These systems, however, lack strong extraction capabilities and fail to use inferred facts together with extracted facts for doing record linkage. Our work with FROntIER strengthens weaknesses in extraction capabilities, adds the ability to infer implied facts of interest, and enables better attribute-based record linkage.

Much more effort has been spent on improving techniques to solve the individual tasks of FROntIER: Sarawagi's book [Sar08] surveys current information extraction techniques. Turmo et al.'s survey of information extraction techniques [TAC06] focuses on statistical methods, and Chang et al.'s survey [CKGS06] compares 19 web information extraction systems. Mishra and Kumar [MK11] survey various semantic web reasoners and languages, and Baader et al.'s handbook [BCM+03] explains the use of inference in description logics. Christen's book [Chr12] surveys techniques for data matching, record linkage, and entity resolution, while Herzog's book [HSW07] focuses on deterministic and probabilistic record linkage techniques. Each of these books and surveys references many dozens of research papers contributing to the three areas spanned by FROntIER's framework.

For FROntIER we select, build on, and synergistically combine this prior work, as follows:

- Our framework extends the capabilities of systems developed by the Data Extraction Research Group at Brigham Young University. Embley et al. [ECJ+99] developed a system, *OntoES*, for ontology-driven extraction with the aid of regular expression based recognizers over HTML pages. Liddle et al. [LHE03] built a development environment for the construction of ontologies called the *Ontology Editor*. Wessman et al. [WLE05] further refined these systems by adding wrappers and facades to facilitate the development of ontologies and the organization of data. We augment this work in FROntIER by developing relationship-based and record-based extractors and by providing object-existence recognizers.
- Our FROntIER framework adds the ability to infer implied facts by adding the Jena reasoner[1], which allows for the construction of inference rules. We use constructed inference rules with the Jena reasoner over extracted facts to organize facts with respect to a target ontology. Our framework also allows for user-defined predicates for use in inference rules by extending the "Builtins" framework provided by Jena.
- Our FROntIER framework includes Duke[2], an off-the-shelf entity-resolution tool, to aid in resolving entities. We create entity-resolution rules for Duke by specifying weights over the various kinds of extracted and inferred facts obtained by FROntIER and generate *owl:sameAs* relationships between entities found in equivalence classes that Duke produces.

Regarding just the information-extraction component of FROntIER, the augmentations developed for this thesis push the state of the art forward. FROntIER's rules are manually specified. Compared with the manual information-extraction systems surveyed in Chen et al.'s work [CKGS06]—TSIMMIS [HMGM97], Minerva [CM98], WebOQL [AM98],

[1] http://jena.apache.org/
[2] http://code.google.com/p/duke/

XWRAP [LPH00], and W4F [SA01]—FROntIER is as strong or stronger in all criteria analyzed: task domain, techniques used, and automation degree. FROntIER's task domain is more challenging as it addresses hand-typeset, OCRed, semi-structured historical documents, which include all of the issues of record variation and attribute granularity normally dealt with in the task domain, plus more. Regarding techniques used, manual extraction systems rely on features such as HTML tags and DOM trees to provide features to guide extraction, but FROntIER must make do without them as it only has OCRed text with which to work. FROntIER's degree of automation is as strong as all the manual systems, but is weaker than extraction systems whose rules are machine-learned. However, none of the 19 extraction systems generates inferred facts, and none resolves object identity as does FROntIER.

Chapter 3

FROntIER

The FROntIER framework has three key components: (1) information extraction with extraction ontologies, (2) inference, and (3) object identity resolution. We discuss the details of each in the succeeding chapters, but as an overview, we first explain how the components fit together to constitute the FROntIER framework.

Figure 3.1 shows how the components in FROntIER are connected and shows the input/output paths of each component. Our target application is historical documents, which are OCRed pages in PDF format. Given a historical document, a user develops an extraction ontology for the document. With the document's pages and the extraction ontology as input, FROntIER invokes OntoES, our **Onto**logy **E**xtraction **S**ystem [ECJ+99], which extracts information from pages of text documents and populates the given ontology with recognized objects, object properties, and relationships between objects and object properties. The output of OntoES is an XML document containing these objects and relationships, which is converted into RDF[1] triples (in an OWL[2] ontology) to be processed by the Jena reasoner. Given a user-specified target ontology and user-developed inference rules, the Jena reasoner produces new implicit facts that comply with the target ontology, which, along with the extracted facts that comply with the target ontology, constitute the populated target ontology. FROntIER outputs the extracted and implicit facts in the target ontology as RDF triples. It also generates a csv (comma-separated value) file by traversing the RDF triples such that each row represents a fact for an entity (for this thesis each entity is a person). Given these

[1] http://www.w3.org/RDF/

[2] http://www.w3.org/TR/owl2-overview/

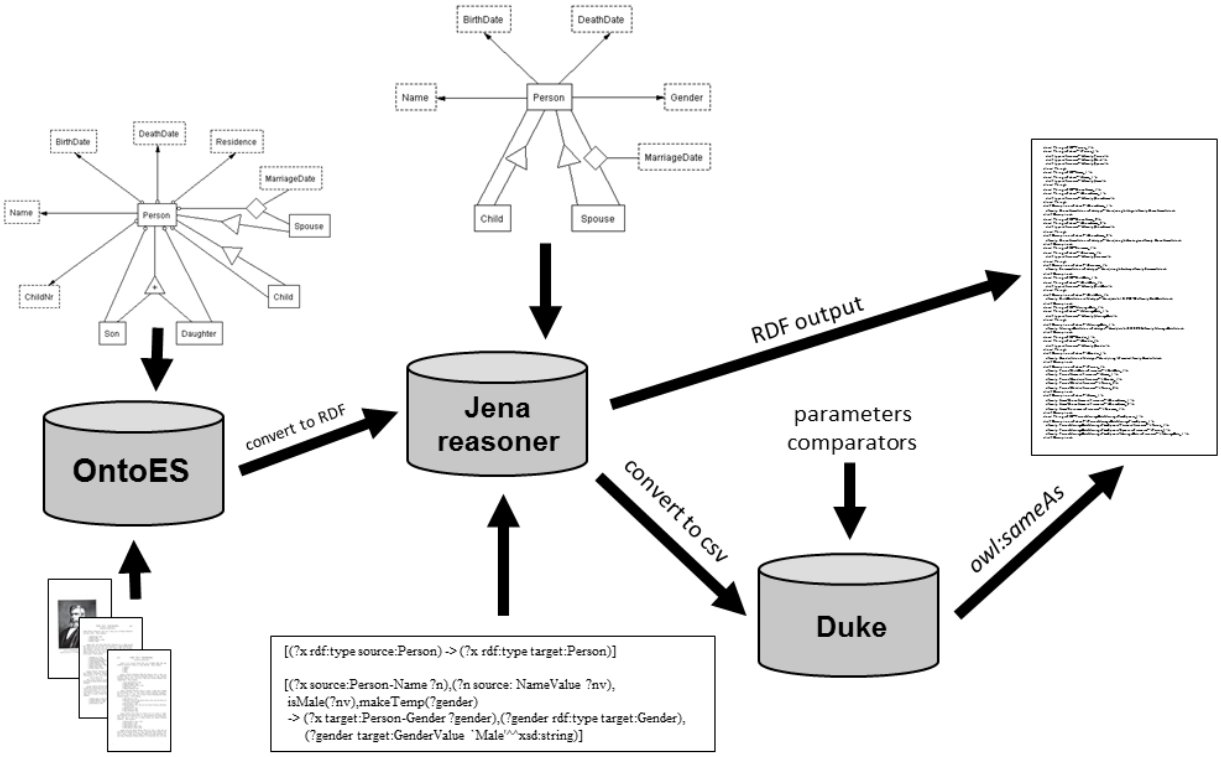

Figure 3.1: Diagram of FROntIER System Architecture.

entity facts and user-specified parameters that weight the various attributes when comparing entity facts, Duke performs identity resolution to disambiguate the entities defined in the csv file. FROntIER produces *owl:sameAs* relationships for Duke-identified coreferent entities, and adds these *owl:sameAs* relationships to the set of RDF triples. These RDF triples are FROntIER's output, which can be queried directly using SPARQL[3], a semantic-web standard for querying RDF triples, or indirectly through HyKSS [ZELS14], a hybrid keyword and semantic search system designed to accommodate free-form and form-based queries over RDF triples.

[3]www.w3.org/TR/rdf-sparql-query

Chapter 4

Extraction Ontologies

An *extraction ontology* is a linguistically grounded conceptual model. Figure 4.1 shows the GUI (Graphical User Interface) of the Ontology Workbench developed previously by the Data Extraction Research Group at BYU. The Ontology Editor is open, displaying the conceptual model diagram of an extraction ontology. The Tools tab is also open, showing access to the tools for linguistically grounding an extraction ontology. As extraction ontologies comprise the first key component of FROntIER, we provide a brief overview of both the conceptual-model component and the linguistic-grounding component. We then proceed to explain the details of the linguistic grounding, the first of the three major contributions of this thesis.

In the conceptual model diagram in Figure 4.1 each box represents an object set. Object sets can either be lexical (represented with dashed lines) or non-lexical (represented with solid lines). Lexical object sets contain strings whereas non-lexical object sets contain surrogates that denote real-world objects. Line segments connecting object sets denote relationship sets, which are usually binary, meaning they only connect two concepts together, but can also be n-ary ($n > 2$). For example, the line segments connecting the *Person*, *MarriageDate*, and *Spouse* object sets in Figure 4.1, which are intersected by a diamond shape, denote a ternary relationship set. Relationship sets can be functional, optional, or both as well as nonfunctional and mandatory. Arrowheads on the range side of relationship sets denote functional relationship sets, and unfilled circles on the domain side denote optional participation of objects in relationships. The absence of arrowheads and unfilled circles

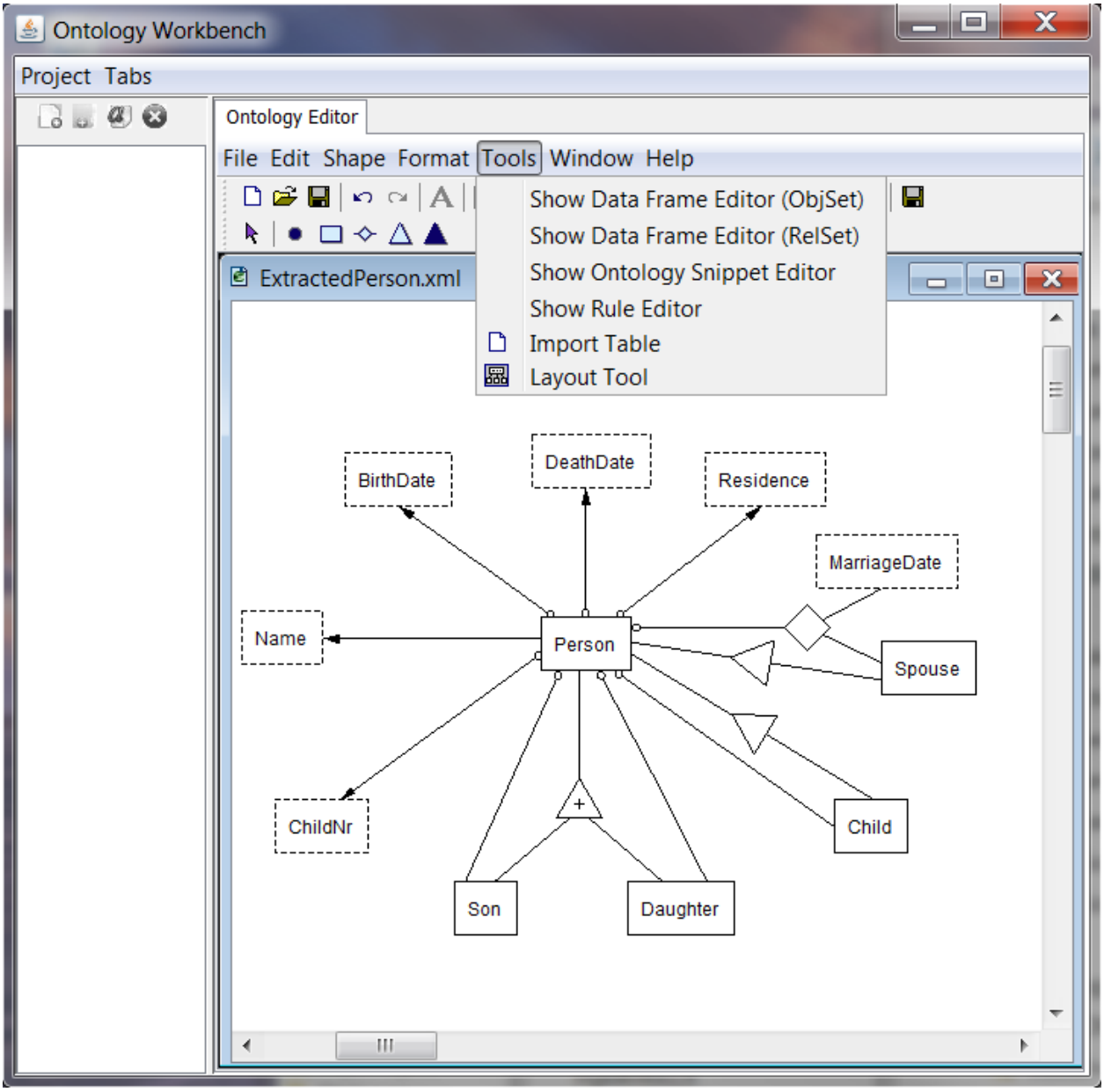

Figure 4.1: The Ontology Editor.

respectively denote nonfunctional relationships and mandatory participation of objects in relationships. An unfilled triangle denotes generalization/specialization with the generalization, or object set that represents the hypernyms, connected to the apex of the triangle and the specializations, or object sets that represent the hyponyms, connected to the base. The set of specializations of a generalization may be disjoint (represented by a '+' symbol as are *Son* and *Daughter* in Figure 4.1) or complete (represented by a '∪' symbol) or both disjoint

and complete, constituting a partition. A filled-in "black" triangle denotes aggregation with the holonym object set, or object set that represents the whole parts, connected to the apex of the triangle and the meronym object sets, or object sets that represent the component parts, connected to the base.

The linguistic component of an extraction ontology consists of four types of instance recognizers—recognizers for lexical object sets, non-lexical object sets, relationship sets, and designated ontology snippets. Instance recognizers are embedded in data frames [Emb80]—abstract data types tied to concepts in an extraction ontology that, in addition to instance recognizers, contain operators that manipulate data values [EZ10]. The Tools menu in Figure 4.1 shows the access to these data-frame definitions: lexical and non-lexical object sets in the first, relationship sets in the second, and ontology snippets in the third. Recognizers for the four types of data frames are similar, but are distinct in some characteristics. We explain each in turn. (Data frames for lexical object sets have been part of OntoES since its inception [ECJ+99]. Data frames for non-lexical object sets, relationship sets, and ontology snippets are part of the development work for this thesis.)

4.1 Lexical Object Sets

Lexical object-set recognizers identify lexical instances in terms of value expressions, context expressions, exception expressions, and dictionaries. Figure 4.2 shows an example of a data-frame recognizer for birth-date years consisting of four-digit year values whose immediate left context is "b. " like all the birth dates in Figure 1.1. Figure 4.3 shows the results of applying the recognizer in Figure 4.2 to Page 419 of *The Ely Ancestry* in Figure 1.1. Careful scrutiny of the displayed values shows that FROntIER correctly extracts all the birth-date years on Page 419 except the birth-date year of Theodore Andruss. Further, scrutiny shows why: the OCR of the birth year for Theodore Andruss is "i860", which is not a four-digit number. It is possible, of course, to allow for this OCR error by letting the the value expression be "\b[i1]\d\d\d\b". Indeed, the left-context exression in Figure 4.2 allows for a comma

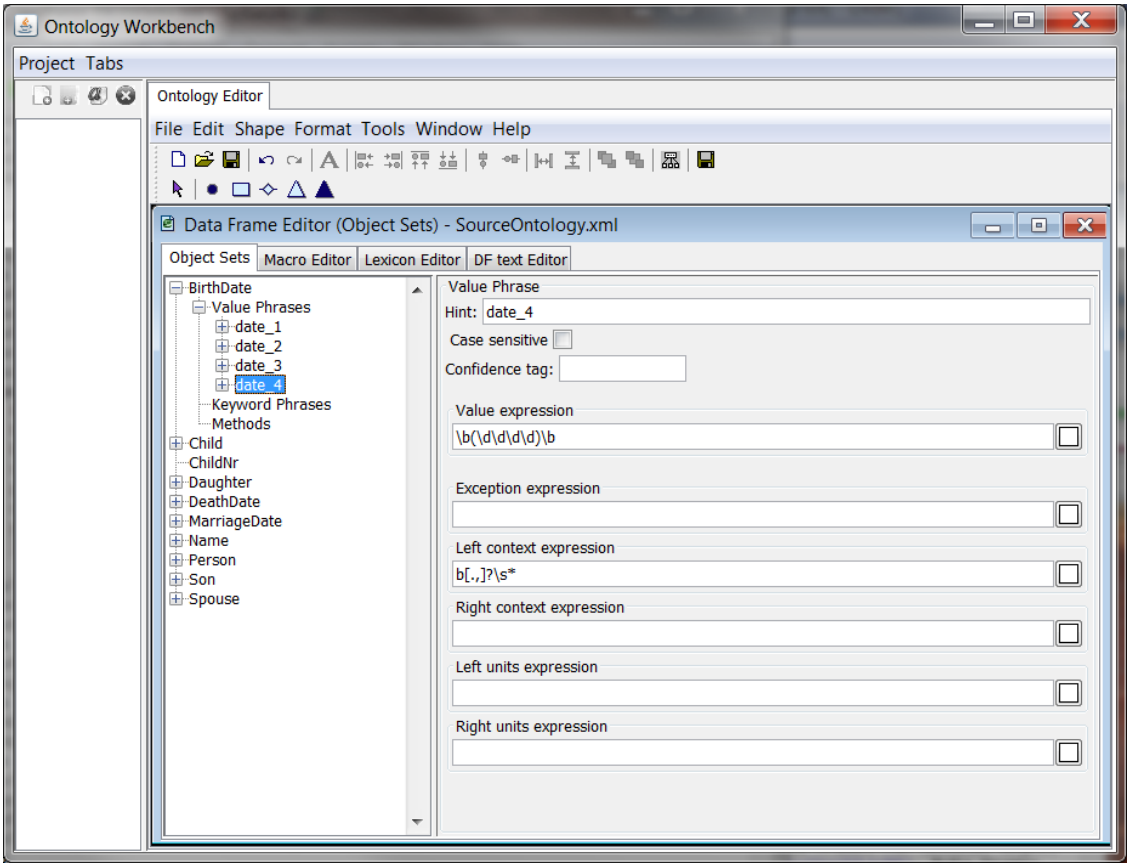

Figure 4.2: A Data-Frame Recognizer for Year Birth Dates.

instead of a period following the "b" which may be caused by an OCR or typesetting error; furthermore, it also allows for missing spaces, extra spaces, or line breaks after "b[.,]", rather than requiring exactly one space.

In general, *value expressions* are regular expressions for specifying how instances may appear in text. *Left context expressions* are regular expressions that match text that must appear immediately before an instance pattern, and likewise, *right context expressions* are regular expressions that match text that must appear immediately after an instance pattern. These context expressions are used to distinguish *BirthDate* values in phrases such as "b. 1836," in Figure 1.1 from *DeathDate* and *MarriageDate* values, whose left contexts are respectively "d. " and "m. ". *Exception expressions* are regular expressions that exclude certain strings that match value expressions. The exception in Figure 4.4, for example,

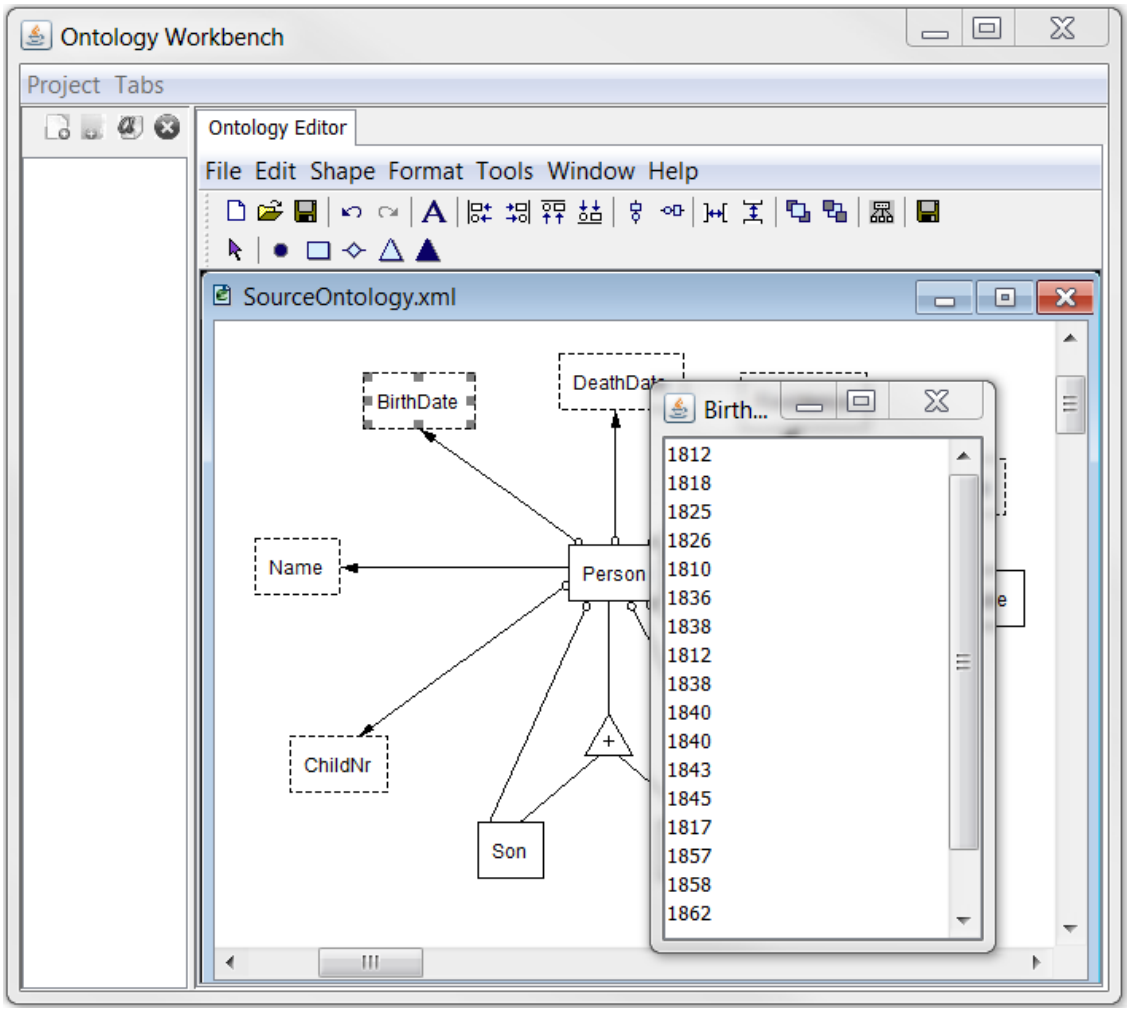

Figure 4.3: Birth-Date Year Results.

excludes illegal dates such as "February 30" that would otherwise match the value expression. Dictionaries are regular expressions where each entry in the dictionary is delimited by an OR ('|'), e.g. for the date recognizer in Figure 4.4, "(January|Jan|February|...)" is part of the *Month* dictionary. Braces around a name—e.g. "{Month}"—refer to a regular expression defined elsewhere.

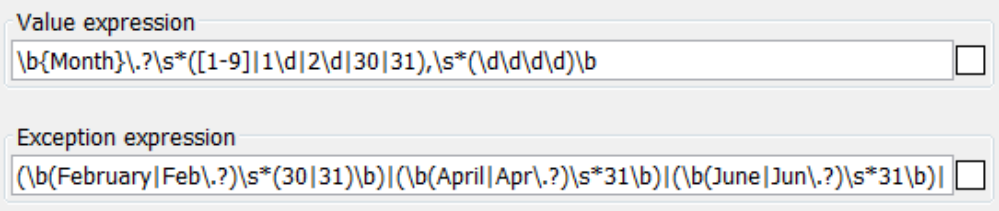

Figure 4.4: Exception Expression and Dictionary Example.

4.2 Non-lexical Object Sets

Non-lexical object-set recognizers identify non-lexical objects through object-existence rules. Object existence rules identify text, such as a proper noun, that designates the existence of an object. An example is a person's name. In Figure 4.5 "{Name}" is the object-existence rule for the *Person* object set. The rule simply references the *Name* object set. When any one of the 19 *Value Phrases* for *Name* in Figure 4.5 recognizes a string of characters as a name, OntoES generates a *Person* object and associates it with the recognized name. Figure 4.6 shows the names and thus the persons extracted from the page in Figure 1.1. Since the *Person* object set is non-lexical, its content is a set of surrogates—object identifiers. Our object identifiers are always "osmx numbers"[1], e.g., "osmx494" for "Mary Eliza Warner" in Figure 4.6. Observe that the object-existence rule populates the two object sets, *Person* and *Name*, as well as the *Person-Name* relationship set.

Object existence rules for non-lexical specializations identify roles for objects in their generalization. The object sets *Son* and *Daughter* in Figure 1.3 are specializations of the *Person* object set and should contain the object identifiers of the respective sons and daughters in *Person*. The object-existence rules in the object sets *Son* and *Daughter* specify which object identifiers in *Person* should also appear *Son* or *Daughter* according to statements made in the document. The object-existence rule in *Son*, for example, is "{Person}[.,]?.{0,50}\s[sS]on\b"

[1]The conceptual-modeling language we use is OSM (Object-oriented Systems Modeling [EKW92]) which is represented internally as XML—hence the "osmx". The appended numbers distinguish objects from one another and are system-generated integers.

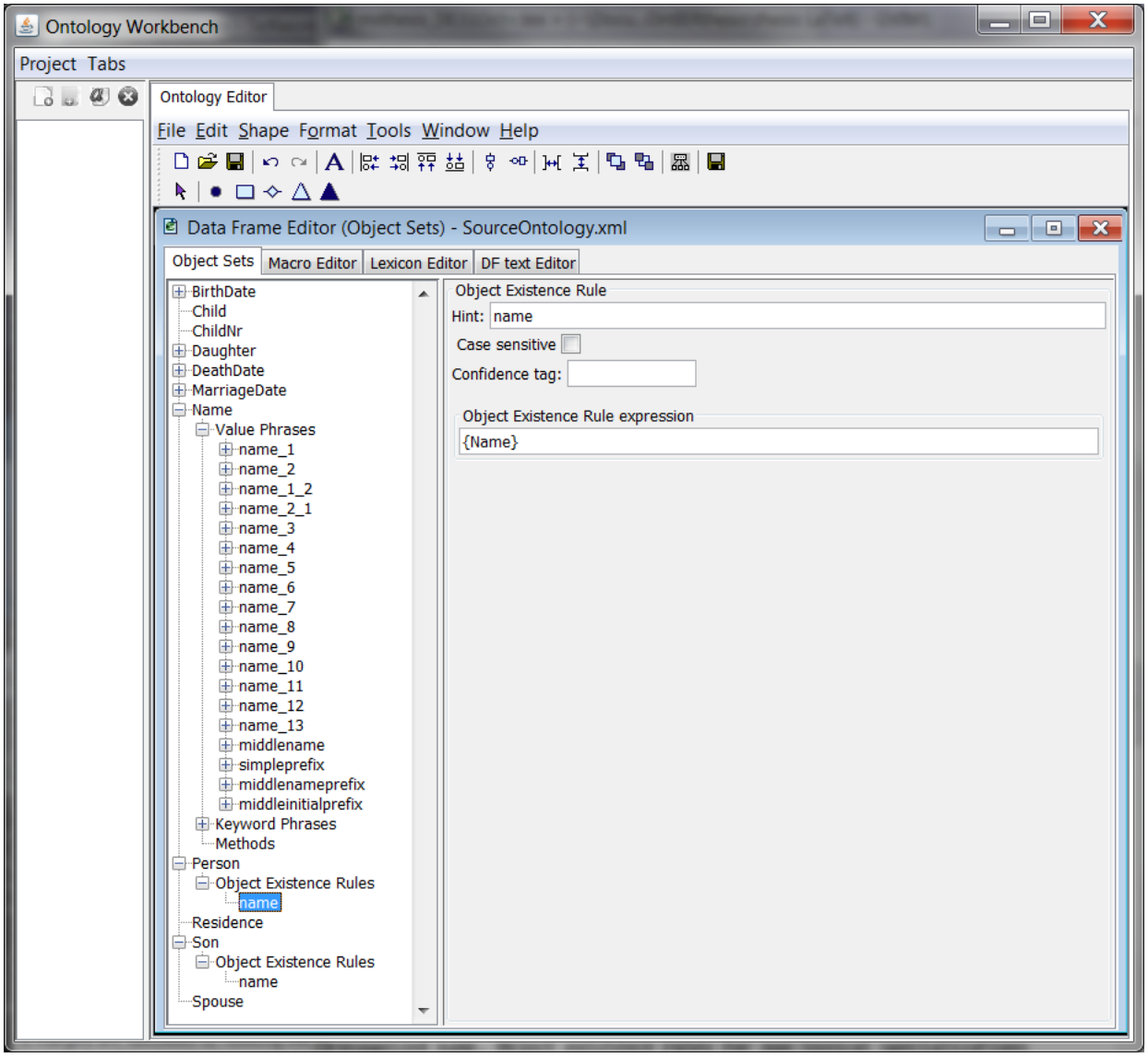

Figure 4.5: Object Existence Rule for the *Person* Object Set.

in Figure 4.7, which references *Person* and establishes the person recognized in the object-existence rule as a son. The rule requires a son to be identified by a name (since "{Name}" is the object-existence rule for *Person*), which must appear before, but not too much before, the word "son" or "Son". Figure 4.8 shows the sons identified in Figure 1.1. It also shows the daughters, which are recognized by a similar rule. The sons and daughters are identified by their surrogate object identifiers. They are a subset of the object identifers in the *Person* object set. Daughter "osmx494" in Figure 4.8 is Person "osmx494" in Figure 4.6, who is

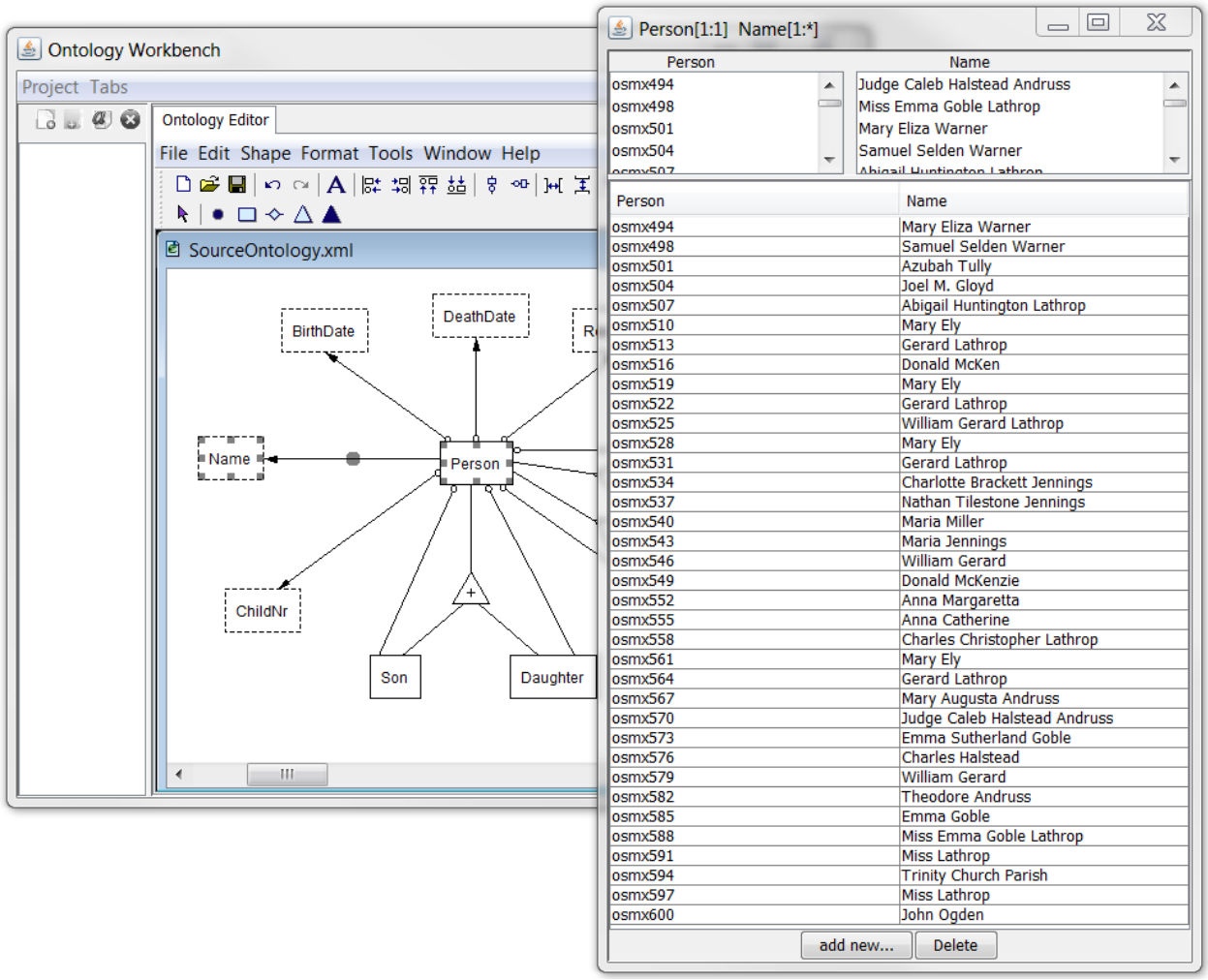

Figure 4.6: Extracted Names and thus also Extracted Persons from the Page in Figure 1.1

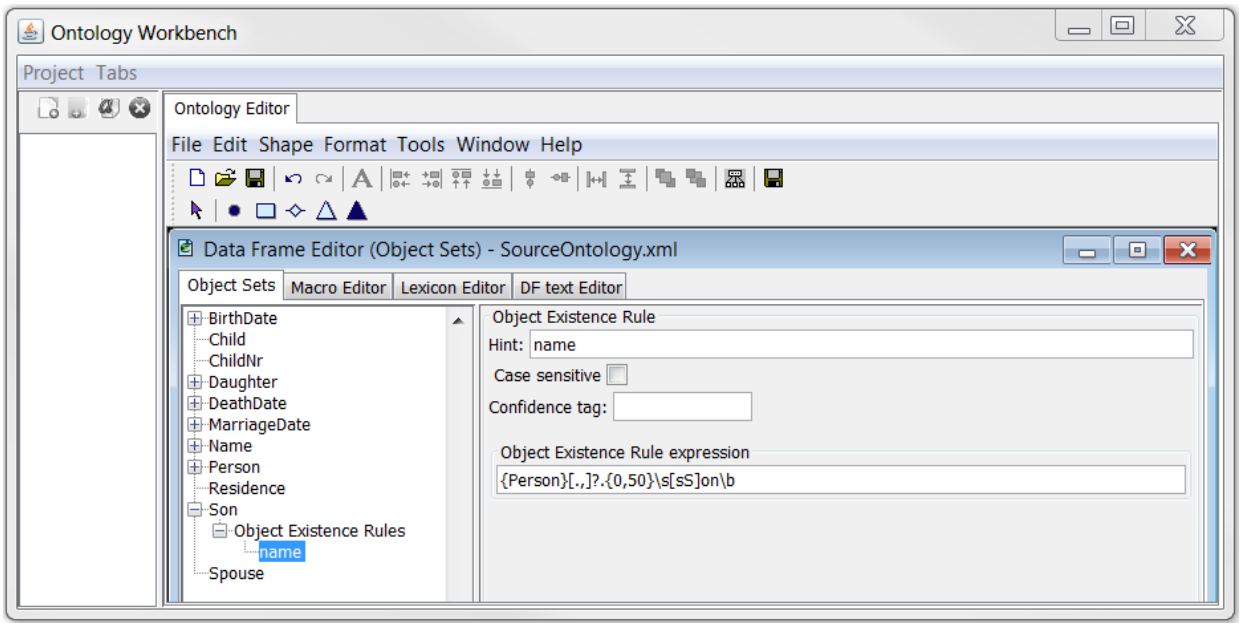

Figure 4.7: Object Existence Rule for the *Person* Object Set.

Mary Eliza Warner, the "dau. of Samuel Selden Warner and Azubah Tully" as stated in Figure 1.1.

Observe that the sons and daughters identified are only those explicitly stated as being sons and daughters in Figure 1.1—the two sons William Gerard Lathrop and Charles Christopher Lathrop and the four daughters, Mary Eliza Warner, Abigail Huntington Lathrop, Charlotte Brackett Jennings, and Mary Augusta Andruss. The other children mentioned in Figure 1.1 are, of course, sons and daughters too, but none is so designated. With FROntIER inference (Section 5) we will be able to determine which of these other children are sons and daughters even though the *Ely Ancestry* page in Figure 1.1 does not so designate them as sons and daughters. This example clearly illustrates the difference between stated and inferred fact assertions.

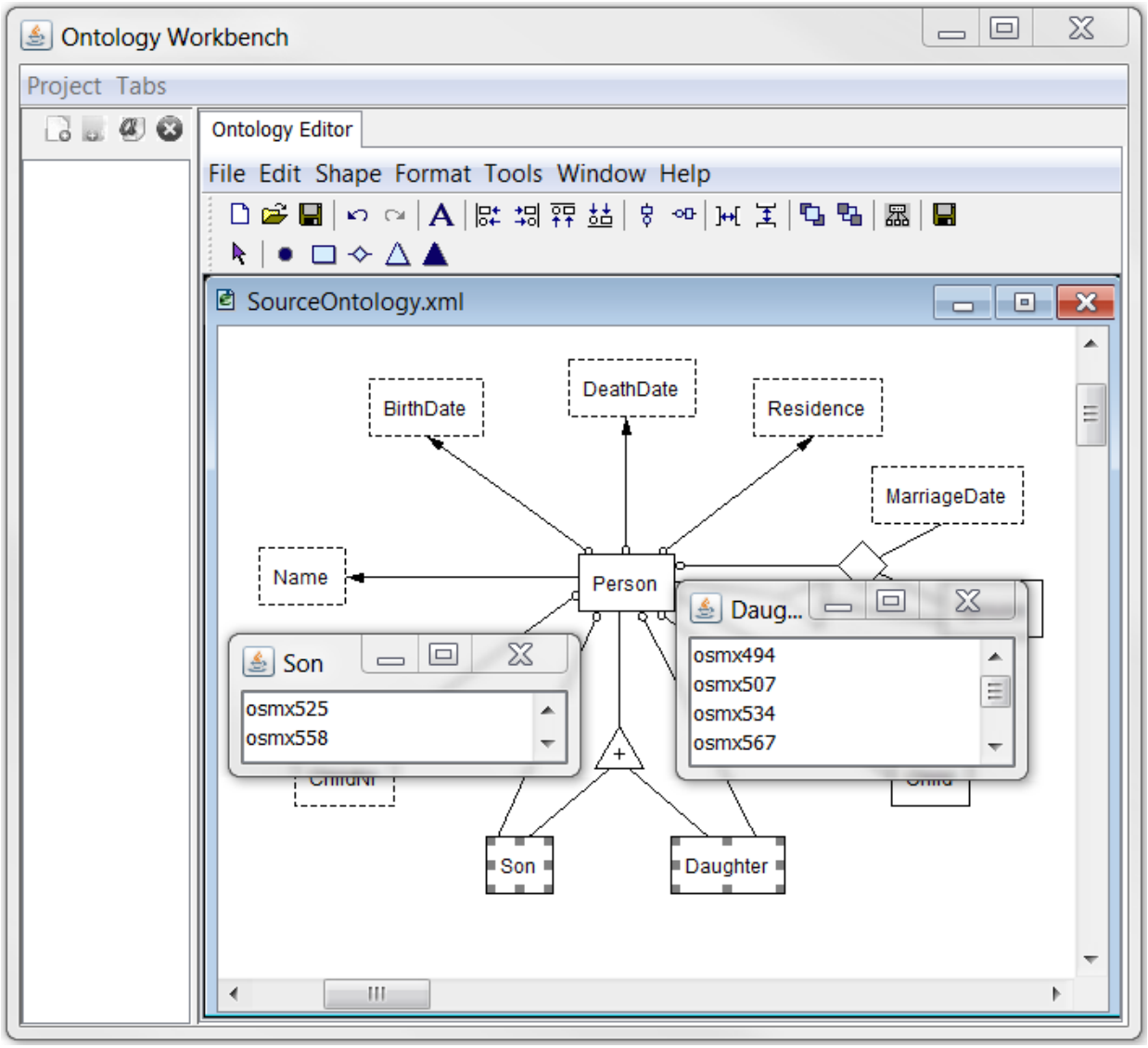

Figure 4.8: Sons Extracted from the Page in Figure 1.3

4.3 Relationship Sets

Relationship-set recognizers identify phrases in a document that relate objects. For example, the *RelPhrase expression* for the *Person-BirthDate* relationship set in Figure 4.9 represents a phrase that relates a person to a birth date. To process the expression, OntoES replaces "{Person}" and "{BirthDate}" with strings previously recognized for the *Person* and *BirthDate* object sets resulting in a regular expression such as "(Maria Jennings|William Gerard|...)[.,]?.{0,10}\s*b[.,]?\s*(1838|1840|...)" which OntoES uses to relate Maria Jennings to 1838 and William Gerard to 1840—two of the *Person-BirthDate* relationships that appear in Figure 1.1. Figure 4.10 shows all the *Person-BirthDate* relationships identified in Figure 1.1 by the rule in Figure 4.9. Several *Person-BirthDate* relationships from Figure 1.1 are missing, such as birth-date years identified by the phrase "who was b." rather than just "b." or with places of residence between the name and birth-date year. These patterns can be picked up with other relationship-recognizer rules. Note, however, that the relationship

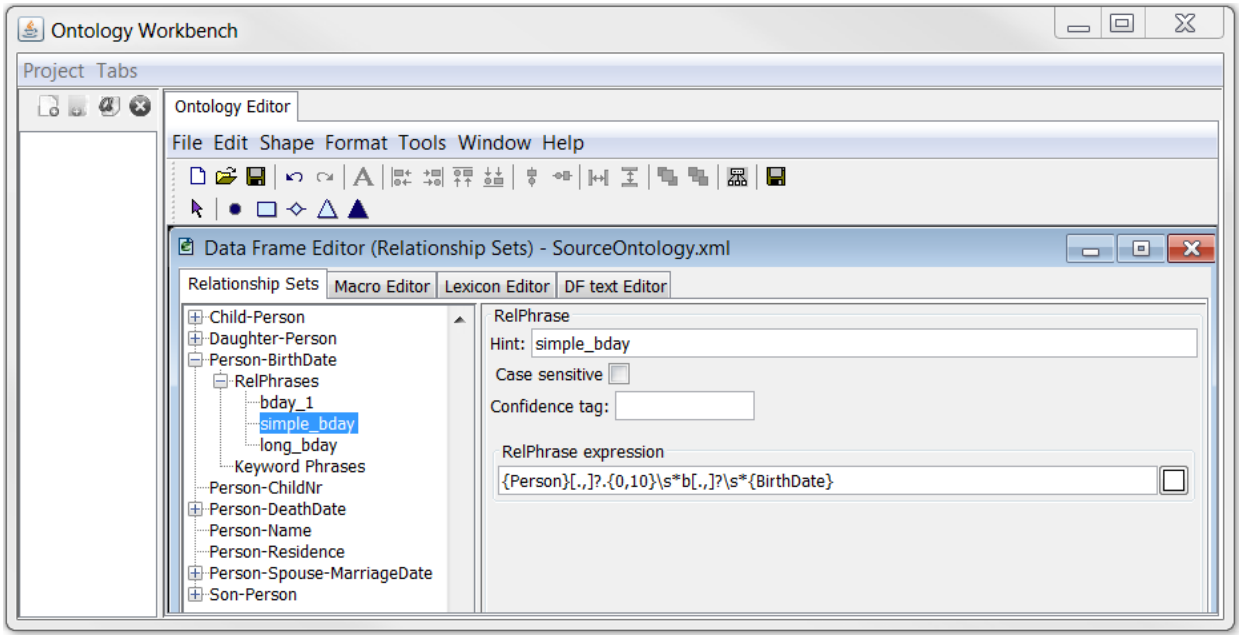

Figure 4.9: *Person-BirthDate* Relationship Set Extraction Rule.

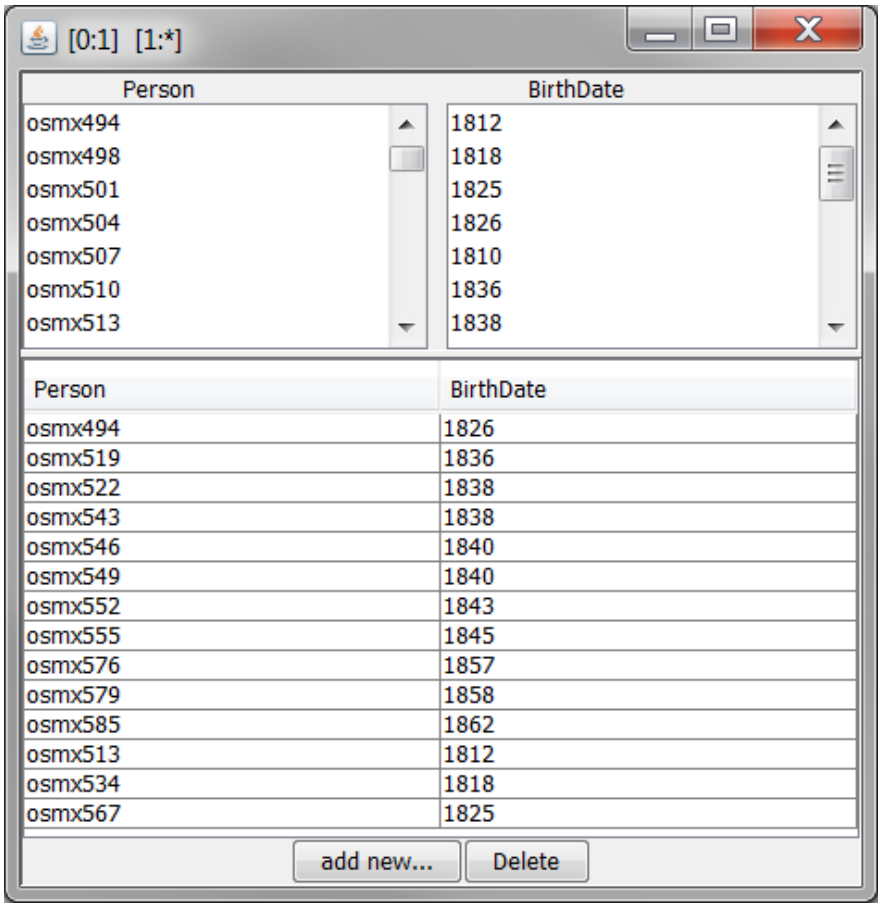

Figure 4.10: Extracted *Person-Birthdate* Relationships.

between Theodore Andruss, *osmx582*, and his birth-date year is missing because of the OCR error, "i860", even though it should be picked up by the relationship recognizer in Figure 4.9.

As another example, the regular expression

```
{Person}[.,]?.{0,50}\s*(son|dau)[.,]?\s+of\s*.{0,50};
\s*m[.,]\s*{MarriageDate}[,]?\s*{Spouse}
```

is the relationship recognizer for the *Person-Spouse-MarriageDate* relationship set. Its results when processed against the page in Figure 1.1 are in Figure 4.11. The regular-expression rule recognizes all four of the stated marriages in Figure 1.1: Mary Eliza Warner and Joel M. Gloyd, Abigail Huntington Lathrop, and Donald McKenzie (although "zie" is missing due to the hyphen-continuation not being fully processed), William Gerard Lathrop and

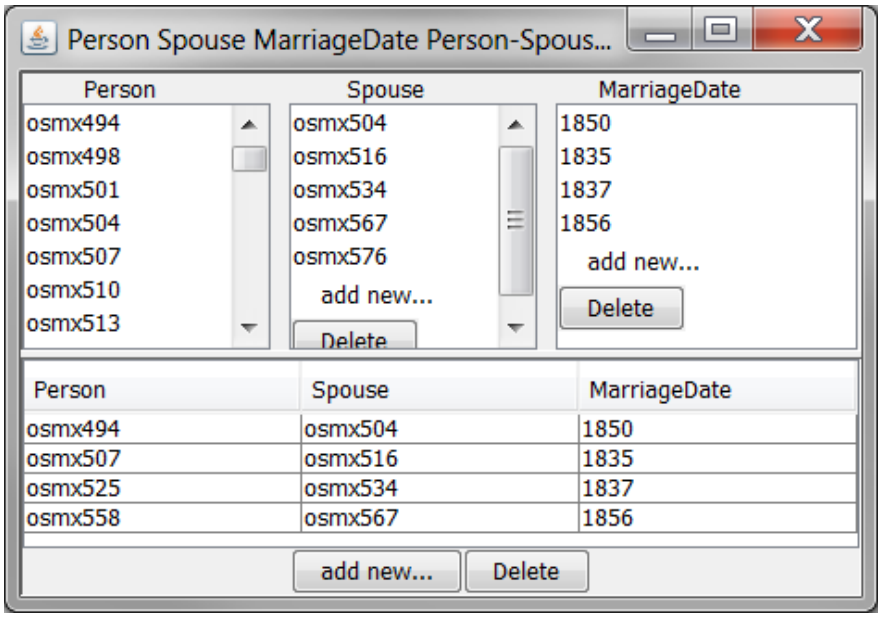

Figure 4.11: Extracted Marriages.

Charlotte Brackett Jennings, and Charles Christopher Lathrop and Mary Augusta Andruss. The implied marriages of the parents of the husbands and wives in these four marriages are not recognized as being stated and must be inferred if the FROntIER system user wishes to recognize the husband-and-wife parent couples as being married.

4.4 Ontology Snippets

Ontology-snippet recognizers extract objects for multiple object and relationship sets as a single unit. Figure 4.12 shows an example. A data frame for ontology snippets consists of an *Ontology Snippet Expression* and *Predicate Mappings*. Ontology snippet expressions are regular expressions with capture groups that map captured instances to ontology predicates—the object and relationship sets in the ontology. Variables for the mappings denote non-lexical objects, and integers denote captured lexical object instances.

The child records in the Ely page in Figure 1.1 show an example of where ontology-snippet recognizers are useful. Each child record comprises a *ChildNr*, *Name*, *BirthDate*, and

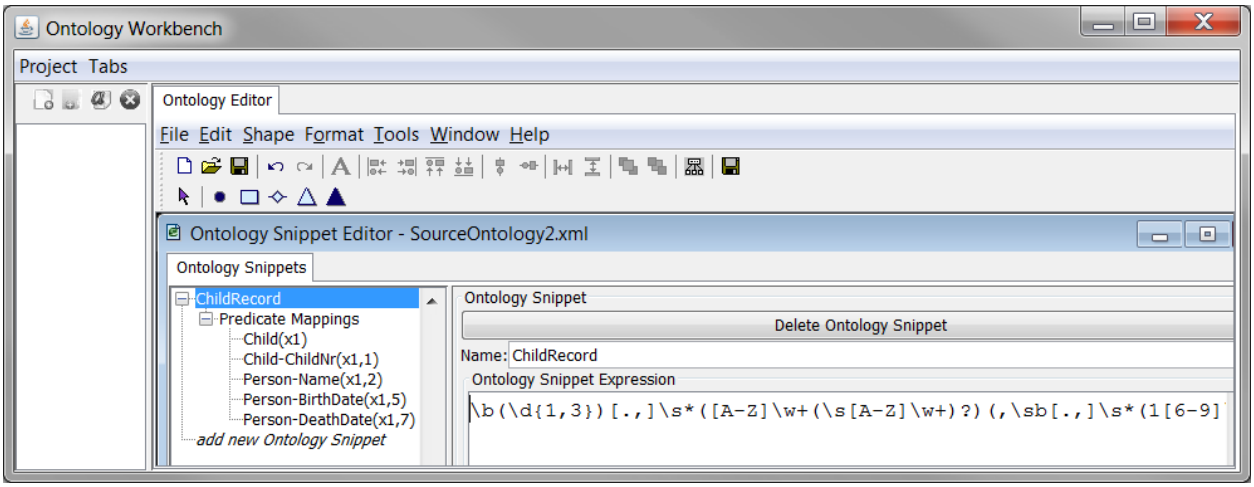

Figure 4.12: Ontology Snippet Declaration.

DeathDate in a particular pattern—an ordered list of record instance values with identical interspersed delimiters. The regular expression in Figure 4.12, which is too long to fit within the window, is:

```
\b(\d{1,3})[.,]\s*([A-Z]\w+(\s[A-Z]\w+)?)(,\sb[.,]\s*(1[6-9]\d\d))?
([,;]\s*d[.,]\s*(1[6-9]\d\d))?[.]
```

Each parenthesized subexpression is a capture group. The first capture group "`(\d{1,3})`" captures the *ChildNr*, and the second "`([A-Z]\w+(\s[A-Z]\w+)?)`" captures the *Name*. The third, fourth, and sixth capture groups do not correspond to lexical instances we wish to capture, but the parenthesized expressions are necessary to properly specify the regular expression. The fifth and seventh capture-group expressions are identical, "`(1[6-9]\d\d)`" and respectively capture the *BirthDate* year and *DeathDate* year. The recognizer, for example, identifies the first two child records in Figure 1.1 as "(1). (Mary (Ely)) (, b. (1836))(, d. (1859))." and "(2). (Gerard (Lathrop)) (, b. (1838))." where the parenthesized expressions represent captured groups numbered left to right by appearance of left parentheses. The predicate mappings specify which substrings map to which object sets, e.g. for the first record: *ChildNr*: *1*, *Name*: *Mary Ely*, *BirthDate*: *1836*, and *DeathDate*: *1859*. The ob-

ject existence rule "`{Name}`" in *Person* and "`\b\d\d?[.]\s{Person}`" in *Child* specify the objects for the non-lexical object sets and provide the connections for the relationship sets.

Applying the ontology-snippet declaration in Figure 4.12 to the page in Figure 1.1 in the context of a second ontology we built yields the results in Figure 4.13. This second ontology has the same conceptual model as the first in Figure 4.1, and thus the ontology snippet is a true sub-component of the ontology diagram comprising only the *Person*, *Child*, *Name*, and *ChildNr* object sets and their interconnections. Further, we included data-frame recognizers only for *Name*, which are needed to support the object existence rules for *Person*

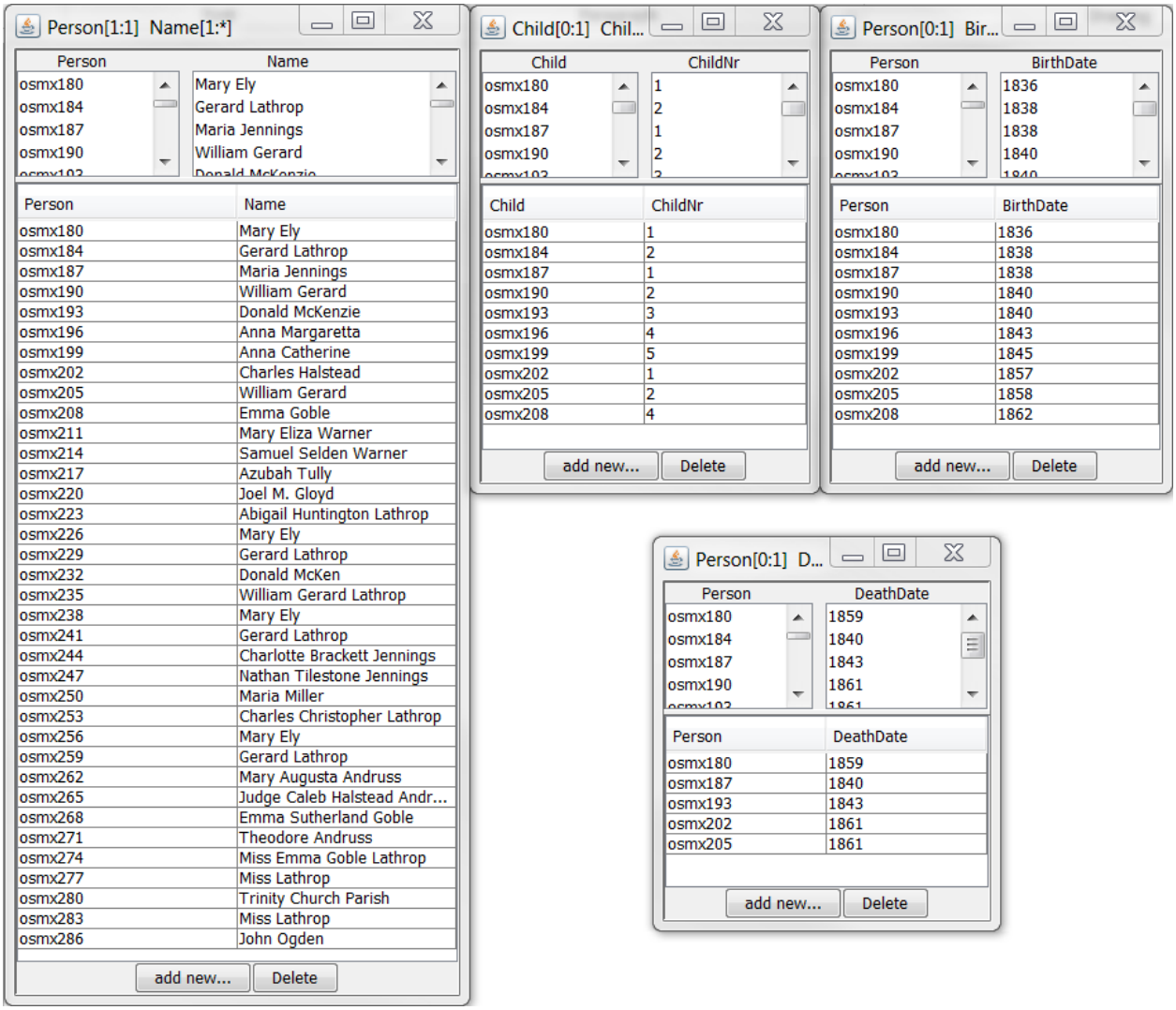

Figure 4.13: Results of Applying the Ontology Snippet Declaration in Figure 4.12

and *Child*. The results in Figure 4.13 show that the ontology-snippet data frame correctly extracted all child records from the page in Figure 1.1, with the exception of the Theodore Andruss record, which has an OCR error in the birth year. The first record, for example, identifies *Mary Ely* as the *Person* with surrogate identifier *osmx180*, which is associated with *ChildNr 1*, *BirthDate 1836*, and *DeathDate 1859*.

Chapter 5

Inference

FROntIER uses rules to organize facts in conformance to a target ontology (e.g. Figure 1.2). Inference is performed using these rules to produce implied facts as well as to transfer or to transform existing facts from a source ontology to a target ontology. As part of the thesis work, a GUI rule editor (shown in Figure 5.1) was created for writing and editing inference rules for FROntIER. We explain the details of how FROntIER uses inference and rules to organize facts in conformance to a target ontology in this chapter.

In order to use the Jena reasoner to do inference, we convert target object and relationship instances into RDF triples. To conform with RDF syntactic requirements, we normalize our ontologies as we convert them. We convert lexical object sets into non-lexicals (RDF classes) with a *Value* property and convert n-ary relationship sets ($n > 2$) into binary relationships connected to a non-lexical (RDF class) that represents the n-ary relationship set. As a result, all relationship sets are binary between two RDF classes, and each lexical object set has a property value associated with its RDF class. Consider for example, the ternary relationship set *Person-MarriageDate-Spouse* in Figure 1.2. The lexical object set *MarriageDate* becomes non-lexical with a *MarriageDateValue* property. We then create a non-lexical object set *PersonMarriageDateSpouse* and form binary relationship sets *PersonMarriageDateSpouse-Person*, *PersonMarriageDateSpouse-MarriageDate*, and *Person-MarriageDateSpouse-Spouse* between the newly created non-lexical object set and the three non-lexical object sets involved in the ternary relationship set. The resulting RDF has four

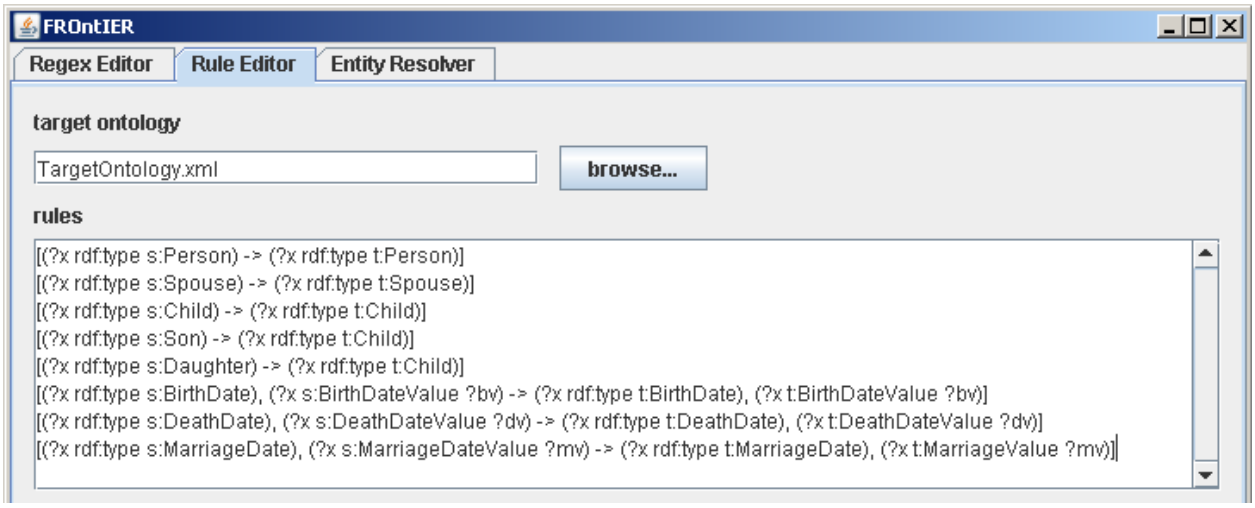

Figure 5.1: GUI for Editing Inference Rules

interconnected classes for these non-lexical object sets, one of which (*MarriageDate*) has a value property.

FROntIER inference rules specify schema mappings between a source ontology s and a target ontology t. The prefix statements:

```
@prefix s: <http://dithers.cs.byu.edu/owl/ontologies/SourceOntology#>.
@prefix t: <http://dithers.cs.byu.edu/owl/ontologies/TargetOntology#>.
@prefix ann: <http://dithers.cs.byu.edu/owl/ontologies/annotation#>.
```

declare name-space identifiers for source and target ontologies and for *ann*, the namespace for the annotations in our extraction ontologies.

Some rules are simply direct transfers of information. The rule:

$$[(?x\ rdf:type\ s:Person)\ \text{->}\ (?x\ rdf:type\ t:Person)] \tag{5.1}$$

creates object identifiers, one for each object identifier in the source object set *Person* and establishes them in the target object set *Person*. The name-space identifiers s and t are bound in the *prefix* statements respectively to the source ontology in Figure 1.3 and the target ontology in Figure 1.2. In the Jena rule syntax, "?x" is a variable (all identifiers preceded by a question mark are variables), and "rdf:type" denotes a class, an object set in our ontologies. In an RDF data store, all data elements are triples. Jena inference rules

work by matching the left-hand side of a rule to triples in the RDF data store; then for every match, the rule generates a triple as specified by the right-hand side of the rule. Thus, for Rule 5.1, every triple specifying that an object *?x* is in the source object set *Person* becomes a triple in the target specifying that the object *?x* is a member of the object set *Person*. Figure 5.2 shows that each object in the source object set *Person* (in the upper-left window of the display of the *Person-Name* relationship set on the left) has become an object in the target object set *Person* (in the upper-left window of the display of the *Person-Name* relationship set on the right) as the two results are in a one-to-one correspondence as can be seen in the lower display windows in Figure 5.2, where we have added the person-names for the object identifiers. (Some of the names in the target ontology have an added surname, which we obtain by inference as discussed later in this chapter.)

As can be seen in Figure 5.2 the *osmx* identifying numbers are *not* the same (e.g. *Mary Eliza Warner* in the source has the identifier *osmx494* while *Mary Eliza Warner* in the target has the identifier *osmx136*). We also point out that although we are showing the results in terms of FROntIER ontologies, all the inferencing actually takes place in the RDF triple store. In addition to a transformation from a populated source ontology to an RDF triple store, FROntIER also has a transformation from a target RDF triple store to a populated target ontology. Thus, the end result of FROntIER inference is a populated target ontology, and hence we show results as populated ontologies.

Rules for birth and death dates as well as for the relationship sets relating persons with their birth and death dates are additional rules that directly transfer information from source to target ontology:

$$[(?x\ rdf{:}type\ s{:}BirthDate),\ (?x\ s{:}BirthDateValue\ ?bv) \\ ->\ (?x\ rdf{:}type\ t{:}BirthDate),\ (?x\ t{:}BirthDateValue\ ?bv)] \tag{5.2}$$

$$[(?x\ rdf{:}type\ s{:}DeathDate),\ (?x\ s{:}DeathDateValue\ ?dv) \\ ->\ (?x\ rdf{:}type\ t{:}DeathDate),\ (?x\ t{:}DeathDateValue\ ?dv)] \tag{5.3}$$

$$[(?x\ s{:}Person\text{-}BirthDate\ ?y)\ ->\ (?x\ t{:}Person\text{-}BirthDate\ ?y)] \tag{5.4}$$

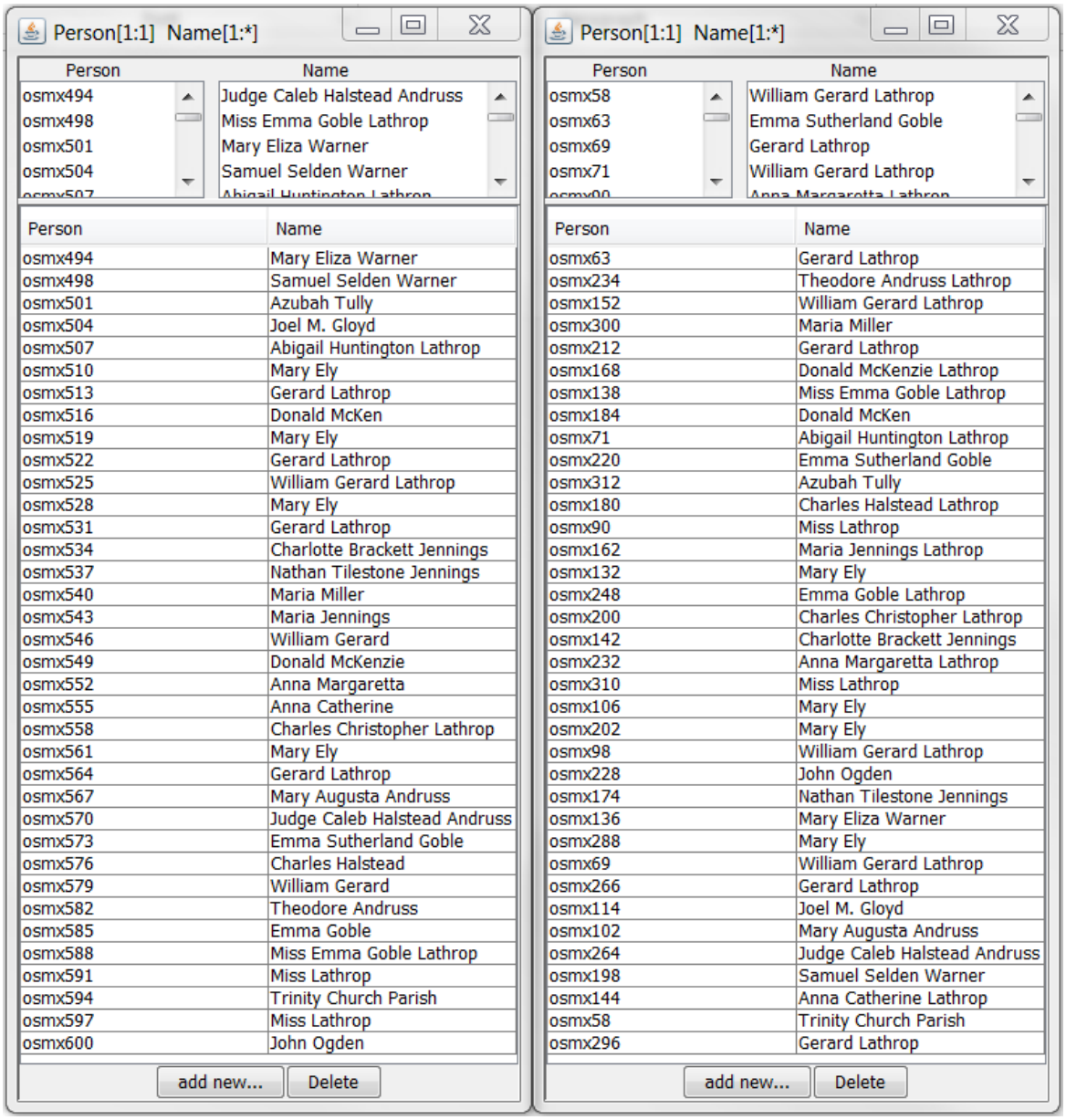

Figure 5.2: Inference Results of Transferring Persons from Source to Target Ontology

$$[(?x\ s{:}\mathit{Person\text{-}DeathDate}\ ?y) \rightarrow\ (?x\ t{:}\mathit{Person\text{-}DeathDate}\ ?y)] \tag{5.5}$$

Observe in Rules 5.2 and 5.3 that the original lexical *BirthDate* and *DeathDate* instances are *BirthDateValue* and *DeathDateValue* instances and are properties of *BirthDate* and *DeathDate* object instances. Figure 5.3 shows the results. For example, in Figure 5.3, *Person osmx142*, who is Charlotte Bracket Jennings as seen in the right-hand list of persons

in Figure 5.2, has birth-date year 1818 as stated in Figure 1.1, and *Person osmx180*, who is Charles Halstead Lathrop has death-date year 1861 as stated in Figure 1.1 where he is Charles Halstead, the son of Charles Christopher Lathrop.

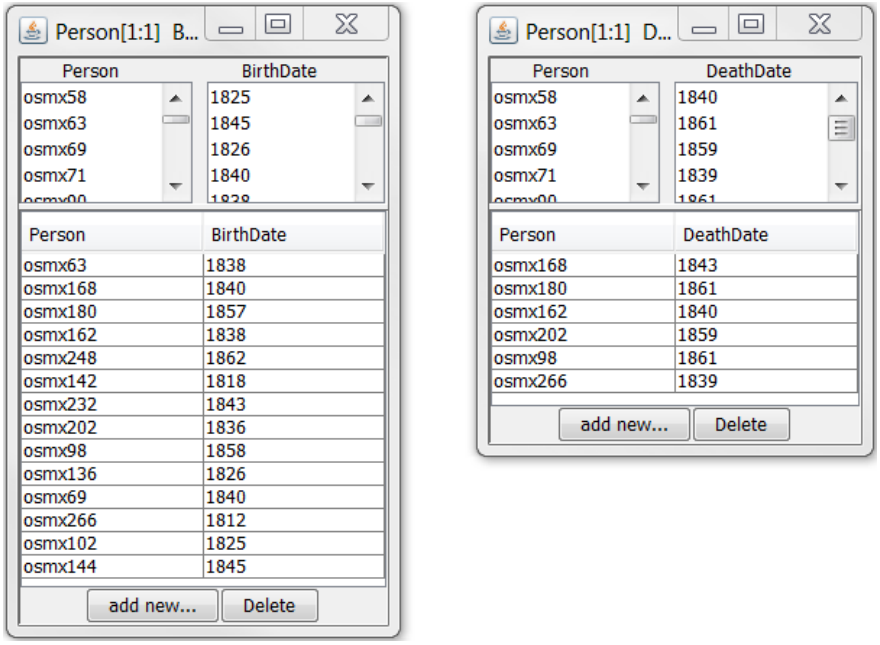

Figure 5.3: Transfer of *BirthDate* and *DeathDate* Information.

Children in the target ontology are obtained from the *Son-Person* and *Daughter-Person* relationship sets as well as from the *Child-Person* relationship sets:

$$[(?x\ rdf{:}type\ s{:}Child) \rightarrow\ (?x\ rdf{:}type\ t{:}Child)] \tag{5.6}$$

$$[(?x\ rdf{:}type\ s{:}Son) \rightarrow\ (?x\ rdf{:}type\ t{:}Child)] \tag{5.7}$$

$$[(?x\ rdf{:}type\ s{:}Daughter) \rightarrow\ (?x\ rdf{:}type\ t{:}Child)] \tag{5.8}$$

$$[(?x\ s{:}Son\text{-}Person\ ?y) \rightarrow\ (?y\ t{:}Person\text{-}Child\ ?x)] \tag{5.9}$$

$$[(?x\ s{:}Daughter\text{-}Person\ ?y) \rightarrow\ (?y\ t{:}Person\text{-}Child\ ?x)] \tag{5.10}$$

$$[(?x\ s{:}Child\text{-}Person\ ?y) \rightarrow\ (?y\ t{:}Person\text{-}Child\ ?x)] \tag{5.11}$$

Figure 5.4 shows the result of executing Rules 5.6–5.11. Observe, for example, that *osmx102* (Mary Augusta Andrus) is a child of *osmx220* (Emma Sutherland Goble), who is her

mother as stated in Figure 1.1. Since *Child* is a specialization of *Person* in the target ontology (Figure 1.2), whenever an object is placed in *Child* it is also automatically placed in *Person*, its generalization. *Generalization/specialization* declarations in an OSM ontology transfer directly to *super-class/sub-class* declarations in RDF/OWL so that populating super-classes happens automatically upon populating sub-classes.

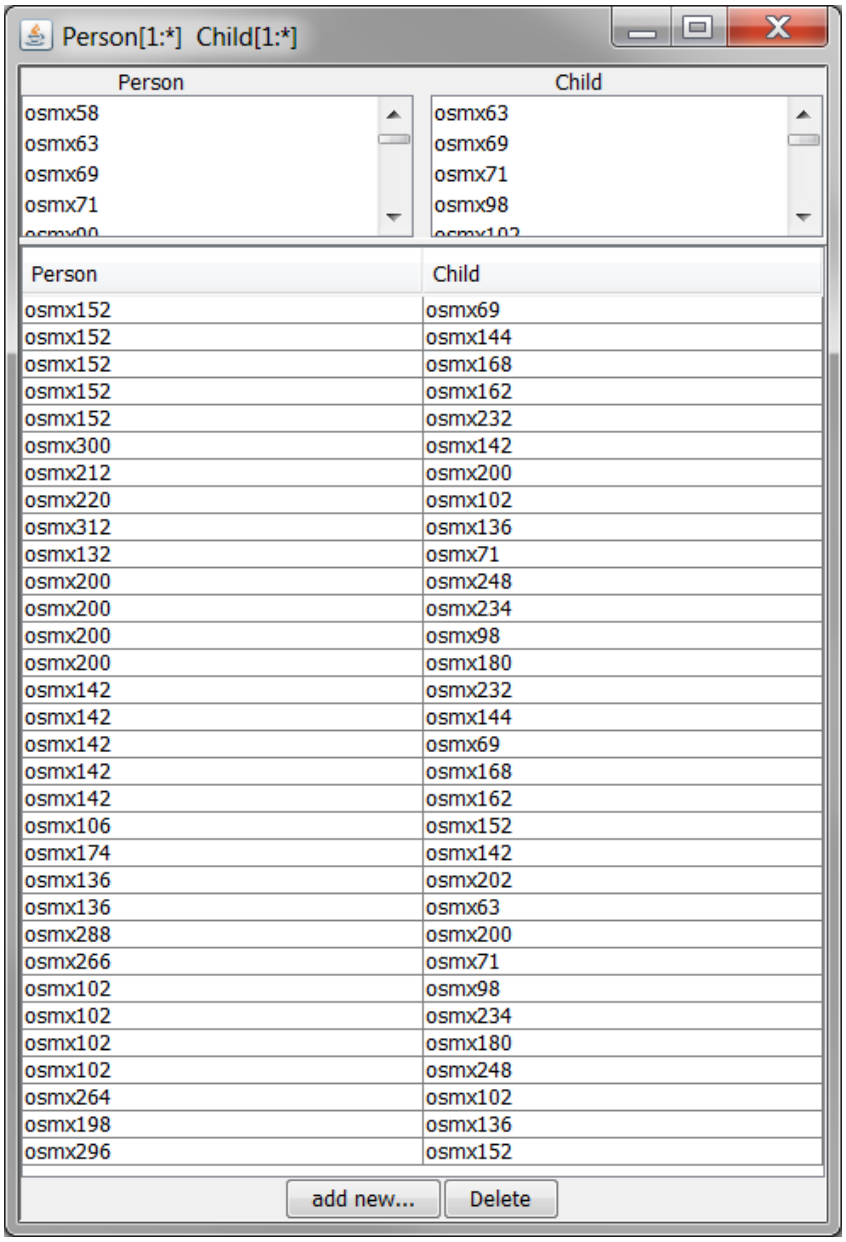

Figure 5.4: Transfer of Parent-Child Information.

The following rules organize both extracted and inferred marriages:

$$[(?x\ rdf{:}type\ s{:}Spouse)\ ->\ (?x\ rdf{:}type\ t{:}Spouse)] \tag{5.12}$$

$$[(?x\ rdf{:}type\ s{:}MarriageDate),\ (?x\ s{:}MarriageDateValue\ ?mv)\ ->\ (?x\ rdf{:}type\ t{:}MarriageDate),\ (?x\ t{:}MarriageValue\ ?mv)] \tag{5.13}$$

$$[(?x\ rdf{:}type\ s{:}PersonSpouseMarriageDate)\ ->\ (?x\ rdf{:}type\ t{:}PersonSpouseMarriageDate)] \tag{5.14}$$

$$[(?x\ s{:}PersonSpouseMarriageDate\text{-}Person\ ?y)\ ->\ (?x\ t{:}PersonSpouseMarriageDate\text{-}Person\ ?y)] \tag{5.15}$$

$$[(?x\ s{:}PersonSpouseMarriageDate\text{-}MarriageDate\ ?y)\ ->\ (?x\ t{:}PersonSpouseMarriageDate\text{-}MarriageDate\ ?y)] \tag{5.16}$$

$$[(?x\ s{:}PersonSpouseMarriageDate\text{-}Spouse\ ?y)\ ->\ (?x\ t{:}PersonSpouseMarriageDate\text{-}Spouse\ ?y)] \tag{5.17}$$

$$\begin{aligned}&[(?x\ s{:}PersonSpouseMarriageDate\text{-}MarriageDate\ ?md),\\&(?x\ s{:}PersonSpouseMarriageDate\text{-}Person\ ?p),\\&(?x\ s{:}PersonSpouseMarriageDate\text{-}Spouse\ ?q),\ notEqual(?p,\ ?q),\\&makeSkolem(?marriageRecord,\ ?p,\ ?q,\ ?md)\\&->\\&(?marriageRecord\ rdf{:}type\ t{:}PersonSpouseMarriageDate),\\&(?marriageRecord\ t{:}PersonSpouseMarriageDate\text{-}Person\ ?q),\\&(?marriageRecord\ t{:}PersonSpouseMarriageDate\text{-}Spouse\ ?p),\\&(?marriageRecord\ t{:}PersonSpouseMarriageDate\text{-}MarriageDate\ ?md)]\end{aligned} \tag{5.18}$$

$$\begin{aligned}&[(?x\ s{:}Son\text{-}Person\ ?a),\ (?x\ s{:}Son\text{-}Person\ ?b),\ notEqual(?a,\ ?b),\\&makeSkolem(?marriageRecord,\ ?a,\ ?b,\ ?x)\\&->\\&(?marriageRecord\ rdf{:}type\ t{:}PersonSpouseMarriageDate),\\&(?marriageRecord\ t{:}PersonSpouseMarriageDate\text{-}Person\ ?a),\\&(?marriageRecord\ t{:}PersonSpouseMarriageDate\text{-}Spouse\ ?b)]\end{aligned} \tag{5.19}$$

$$\begin{aligned}&[(?x\ s{:}Daughter\text{-}Person\ ?a),\ (?x\ s{:}Daughter\text{-}Person\ ?b),\ notEqual(?a,\ ?b),\\&makeSkolem(?marriageRecord,\ ?a,\ ?b,\ ?x)\\&->\\&(?marriageRecord\ rdf{:}type\ t{:}PersonSpouseMarriageDate),\\&(?marriageRecord\ t{:}PersonSpouseMarriageDate\text{-}Person\ ?a),\\&(?marriageRecord\ t{:}PersonSpouseMarriageDate\text{-}Spouse\ ?b)]\end{aligned} \tag{5.20}$$

Rules 5.12–5.17 copy the basic marriage information from source ontology to target ontology. Rule 5.18 infers the symmetric relationship of spouses as an additional marriage instance. The extracted facts only consider the second person in the relationship as a spouse and not the first person. Rule 5.18 finds a *Person ?q* married to a *Spouse ?p*, makes a new surrogate-object marriage instance for the *PersonSpouseMarriageDate* class with the built-in *makeSkolem* predicate, and builds the relationships with *?q* and *?p* switched. Rules 5.19 and 5.20 infer the existence of marriage relationships of parents. Rule 5.19 finds the two parent objects *?a* and *?b* for a son *?x*, makes a surrogate object for the marriage, and attaches the marriage information. Rule 5.20 for daughters is similar. The results are in Figure 5.5. As an example of establishing the converse *Person-Spouse* relationship based on symmetry, the marriage between William Gerard Lathrop (*osmx152*) and Charlotte Bracket Jennings (*osmx142*) appears both as (*Person:osmx152*, *Spouse:osmx142*) and as (*Person:osmx142*, *Spouse:osmx152*). As an example of an inferred marriage that was not extracted in the source, observe in Figure 1.1 that Charlotte's parents, Nathan Tilestone Jennings (*osmx174*) and Maria Miller (*osmx300*) are listed in Figure 5.5 as being married—indeed listed twice, once with Nathan Tilestone Jennings as the *Spouse* and once with Maria Miller as the *Spouse*.

Rules 5.21–5.24 infer the gender of a person:

$$\begin{array}{l}[(?x\ rdf{:}type\ s{:}Son),\ makeSkolem(?gender,\ ?x)\\ ->\\ (?x\ t{:}Person\text{-}Gender\ ?gender),\ (?gender\ rdf{:}type\ t{:}Gender),\\ (?gender\ t{:}GenderValue\ 'Male')]\end{array} \tag{5.21}$$

$$\begin{array}{l}[(?x\ rdf{:}type\ s{:}Daughter),\ makeSkolem(?gender,\ ?x)\\ ->\\ (?x\ t{:}Person\text{-}Gender\ ?gender),\ (?gender\ rdf{:}type\ t{:}Gender),\\ (?gender\ t{:}GenderValue\ 'Female')]\end{array} \tag{5.22}$$

$$\begin{array}{l}[(?x\ s{:}Person\text{-}Name\ ?n),\ (?n\ rdf{:}type\ s{:}Name),\ (?n\ s{:}NameValue\ ?nv),\\ noValue(?x\ rdf{:}type\ s{:}Son),\ noValue(?x\ rdf{:}type\ s{:}Daughter),\\ isMale(?nv),\ makeSkolem(?gender,\ ?x)\\ ->\\ (?x\ t{:}Person\text{-}Gender\ ?gender),\ (?gender\ rdf{:}type\ t{:}Gender),\\ (?gender\ t{:}GenderValue\ 'Male')]\end{array} \tag{5.23}$$

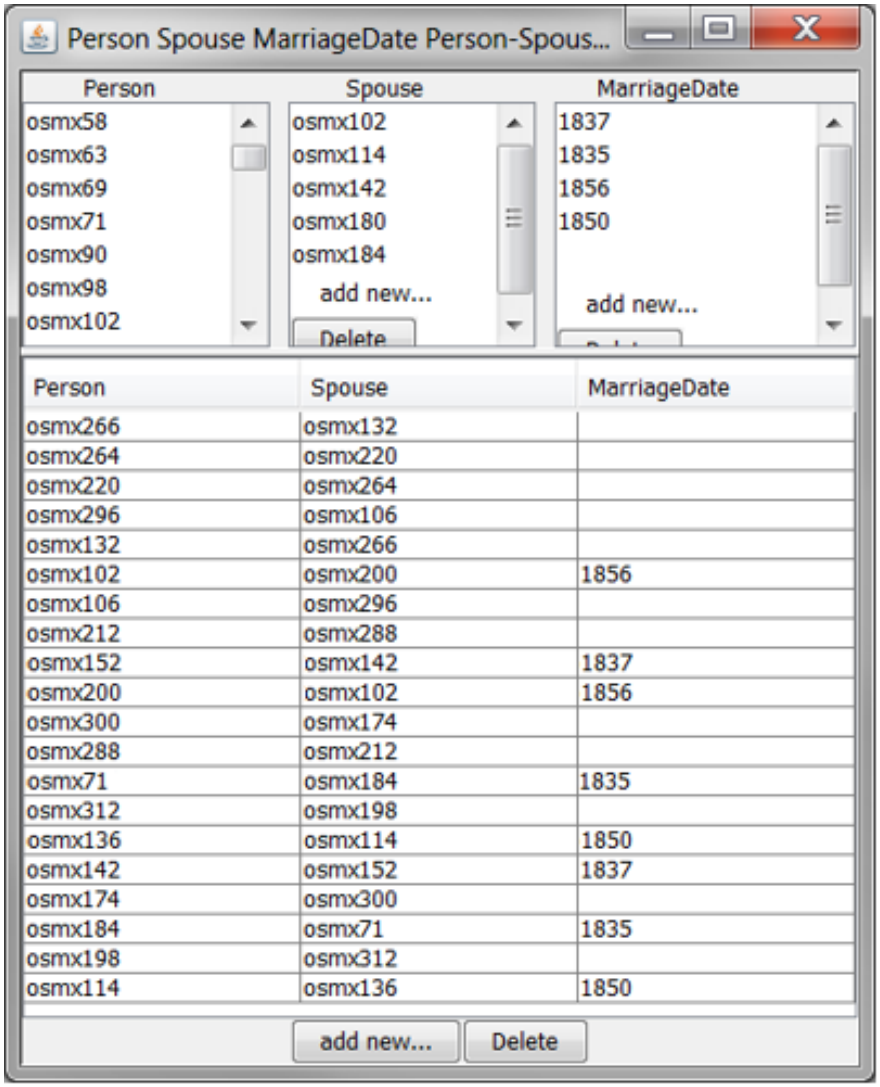

Figure 5.5: Transfer and Inference of Marriage Information.

$$
\begin{aligned}
&[(?x\ s{:}Person\text{-}Name\ ?n),\ (?n\ rdf{:}type\ s{:}Name),\ (?n\ s{:}NameValue\ ?nv),\\
&noValue(?x\ rdf{:}type\ s{:}Son),\ noValue(?x\ rdf{:}type\ s{:}Daughter),\\
&isFemale(?nv),\ makeSkolem(?gender,\ ?x)\\
&->\\
&(?x\ t{:}Person\text{-}Gender\ ?gender),\ (?gender\ rdf{:}type\ t{:}Gender),\\
&(?gender\ t{:}GenderValue\ 'Female')]
\end{aligned}
\tag{5.24}
$$

Rules 5.21 and 5.22 infer gender from stated son and daughter relationships whereas, for those not so designated and sons or daughters, Rules 5.23 and 5.24 infer gender from given names. Our inference rules are constrained to the set of constructs supported by

the Jena framework. Conveniently, the Jena framework defines a set of built-in predicates that is extendable. For extending the set of built-ins, the Jena framework provides the implementation of a *builtin* interface, and we implement this interface for each user-defined built-in. In Rules 5.23 and 5.24, for example, we use the user-defined built-ins *isMale* and *isFemale*, which access a predefined statistical table [Sch12] to determine whether a name is for a male or a female. Figure 5.6 shows that every person in the target ontology has been given the correct gender. Note that the spurious name "Trinity Church Parish" (*osmx58*) has not been assigned a gender: the name is neither designated as the name of a son nor as a daughter in the document page, and it does not appear with high enough probability as being either a male name or female name in the statistical table.

Rule 5.25 infers birth surnames for the children listed in Figure 1.1:

$$
\begin{array}{l}
[(?c\ t{:}\mathit{Person\text{-}Child}\ ?p),\ (?p\ s{:}\mathit{Person\text{-}Name}\ ?n),\ (?n\ s{:}\mathit{NameValue}\ ?nv),\\
(?p\ t{:}\mathit{Person\text{-}Gender}\ ?g),\ (?g\ t{:}\mathit{GenderValue}\ 'Male'),\ (?c\ \mathit{rdf}{:}\mathit{type}\ s{:}\mathit{Child}),\\
(?c\ s{:}\mathit{Person\text{-}Name}\ ?cn),\ (?cn\ s{:}\mathit{NameValue}\ ?cnv),\ (?cn\ \mathit{ann}{:}\mathit{Annotation}\ ?a),\\
\mathit{noValue}(?c\ \mathit{rdf}{:}\mathit{type}\ s{:}\mathit{Son}),\ \mathit{noValue}(?c\ \mathit{rdf}{:}\mathit{type}\ s{:}\mathit{Daughter}),\\
\mathit{getsurname}(?nv,\ '(([A\text{-}Z][A\text{-}Z\,a\text{-}Z]+)[\text{-}\ ]*)+',\ ?x),\ \mathit{strConcat}(?cnv,\ '\ ',\ ?x,\ ?nx)\\
->\\
(?cn\ \mathit{rdf}{:}\mathit{type}\ t{:}\mathit{Name}),\ (?cn\ t{:}\mathit{NameValue}\ ?nx),\ (?a\ \mathit{ann}{:}\mathit{DisplayValue}\ ?nx),\\
(?a\ \mathit{ann}{:}\mathit{CanonicalValue}\ ?nx),\ \mathit{remove}(7)]
\end{array}
\tag{5.25}
$$

Rule 5.25 states that if *?c* is the child of *?p* whose gender is 'Male' (i.e. *?p* is the father of *?c*), then both the annotation (*ann*) for the *DisplayValue* and the *CanonicalValue* of the child's name *?cn* is *?nx*, which is the string-concatenation (*strConcat*) of the name of the child *?cnv* and the surname of the father *?x*, obtained by parsing out the last name of the father with the regular expression in the user-defined predicate *getsurname*. (In Rule 5.25 *remove(7)* refers to the 7th predicate on the left-hand-side of the rule, starting with a 0 count, which specifies that the child's *NameValue* is to be removed so that a new one can be assigned.) As Figure 5.2 shows, for example, "Maria Jennings" in the source ontology (on the left-hand side) becomes "Maria Jennings Lathrop" in the target ontology (on the right-hand side). Note that the erroneously extracted surname "McKen" is not attached to

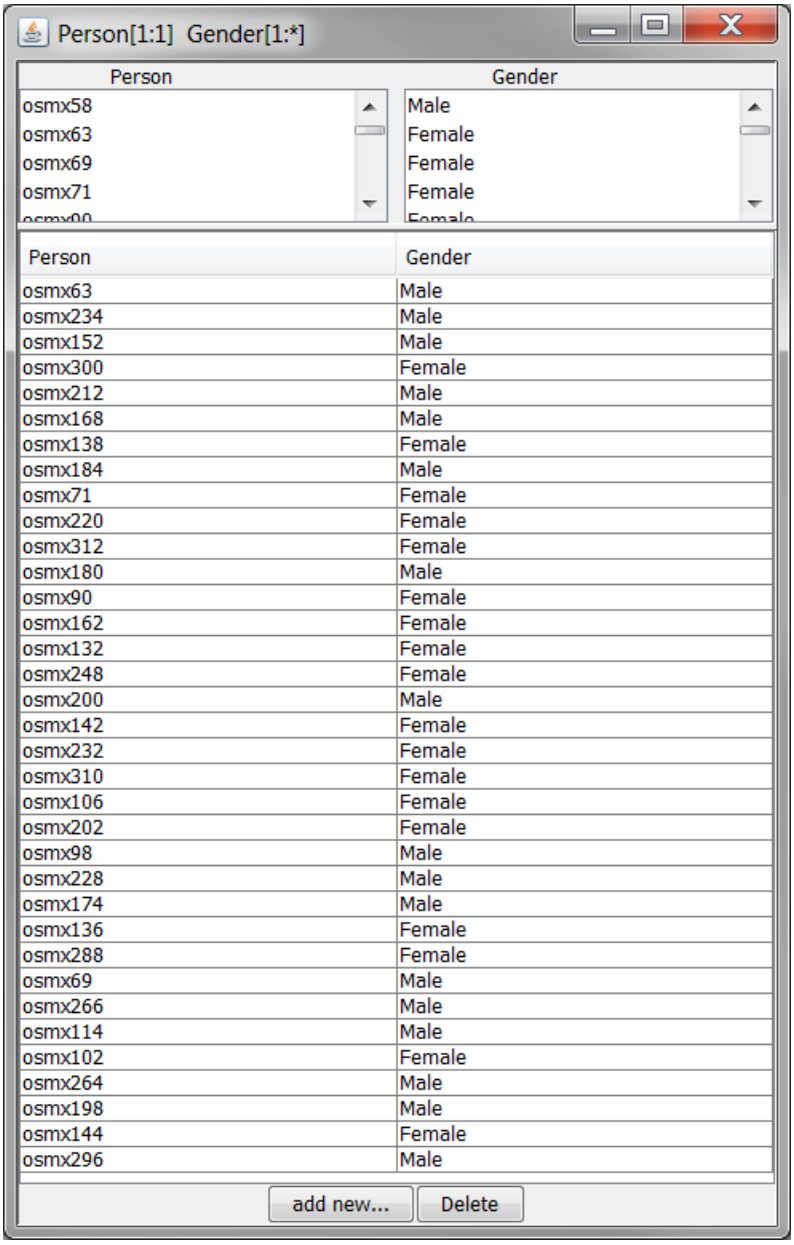

Figure 5.6: Gender Results.

Donald McKenzie's children, Mary Ely and Gerard Lathrop as the page in Figure 1.1 implies they should have been. This is because of an original extraction error: Mary and Gerard are not recognized as being the children of Donald.

Chapter 6

Object Identity Resolution

Object identity resolution in FROntIER is about determining whether any two object identifiers in the *Person* object set designate the same person. Object-existence rules make a new surrogate identifier for every extracted *Name*. In Figure 1.1, for example, Mary Ely the mother of Abigail Huntington Lathrop and Mary Ely the mother of William Gerard Lathrop are the same person, but their generated surrogate identifiers are different, respectively *osmx510* and *osmx528* as Figure 4.6 shows.

FROntIER's object identity resolution uses facts for entities in populated target ontologies as input and generates *owl:sameAs* relationships as output. FROntIER can use any off-the-shelf or specially developed fact-based entity resolver. For our thesis work we used Duke[1], an off-the-shelf entity resolver.

In order to use Duke, we convert the inferred RDF triples output by Jena into a csv file, which can be viewed as a table of entity records. The conversion from RDF triples to csv records is hand-specified—once for each target ontology within a domain. For the target ontology in Figure 1.2, we produce csv records as follows: convert non-lexicals with a *Value* property into table attributes such as *BirthDate* with a *BirthDateValue* property into the attribute *BirthDate*; convert the ternary relationship set *Person-MarriageDate-Spouse* into *MarriageDate* and *Spouse* attributes, where the non-lexical specialization *Spouse* becomes *SpouseName* through its generalization's object existence rule; and calculate the maximum observed cardinality for *Person-Child* instances and for *Person-MarriageDate-Spouse* instances to produce the attributes *Child1Name*, *Child2Name*, ..., *Spouse1Name*, ...,

[1] http://code.google.com/p/duke/

MarriageDate1, ... up to the maximum number of instances for each. Figure 6.1 shows some of the records created from the inference-rule populated target ontology—the target data in Figures 5.2–5.6. The first line of the csv file specifies the attributes.

```
Person,Name,BirthDate,DeathDate,Gender,Spouse1Name,MarriageDate1,Child1Name,
        Child2Name,Child3Name,Child4Name,Child5Name
osmx132,Mary Ely,,,Female,Gerard Lathrop,,Abigail Huntington Lathrop,,,,,
osmx202,Mary Ely,1836,1859,Female,,,,,,,,
osmx106,Mary Ely,,,Female,Gerard Lathrop,,William Gerard Lathrop,,,,,
osmx266,Gerard Lathrop,,,Male,Mary Ely,,,,,,,
osmx63,Gerard Lathrop,1838,,Male,,,,,,,,
osmx152,William Gerard Lathrop,1812,1882,Male,Charlotte Bracket Jennings,1837,
        Maria Jennings Lathrop,William Gerard Lathrop,Donald McKenzie Lathrop,
        Anna Margaretta Lathrop,Anna Catherine Lathrop,
osmx296,Gerard Lathrop,,,Male,Mary Ely,,,,,,,
osmx69,William Gerard Lathrop,1840,,Male,,,,,,,,
```

Figure 6.1: Comma-Separated Value (csv) File of Some of the Persons in Figure 1.1.

The Duke entity resolver uses a configuration file to set attribute comparators and parameter values. For our thesis work we used the ExactComparator for all attributes (which matches two attributes only if their string values are identical). For parameter values, each attribute has a low value for when two attribute-value pairs do not match and a high value for when they do match. Duke combines the values to produce a probability that two entities are the same. Figure 6.2 shows the interface we built for FROntIER to set these parameters. As the figure shows, we set the high value to 0.73 for birth-date years because we believe that when they match they are moderately discriminating, and we set the low value to 0.0020 because mismatched birth-date years are highly discriminating. As additional examples, since matching names indicate but do not guarantee that two people are the same, we set the high value for name match to 0.60. We set the low value for name mismatch to 0.45 since having different names does not mean that the people to which the names refer are different (e.g. in Figure 1.1 Mary Augusta Andruss is also referred to as Mrs. Lathrop). Gender does not disambiguate persons when they match but is very discriminating when they do

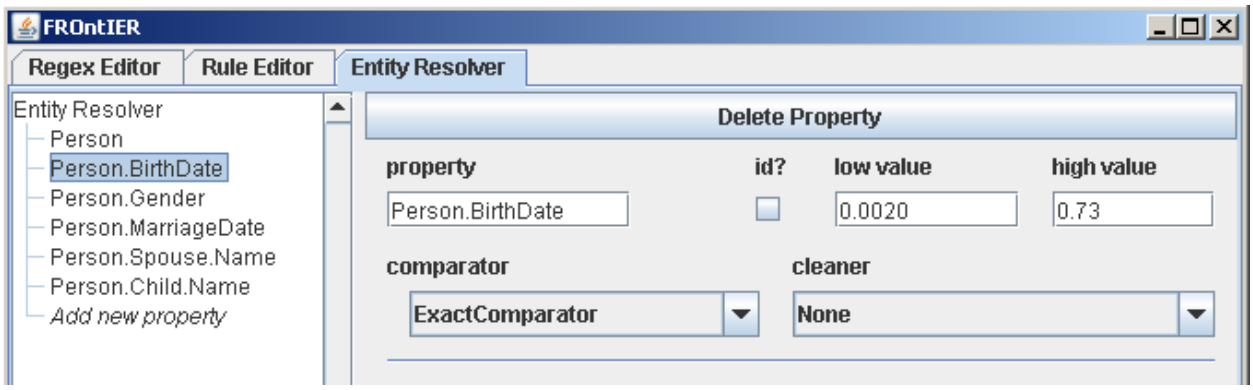

Figure 6.2: Parameter Setting for Object Identity Resolution

not match, so a high value of 0.56 and a low value of 0.01 are appropriate. Similarly, we set other parameter values according to expected significance within the domain.

After running Duke over a file that includes the records in Figure 6.1, it concludes that the probability that the first and third Mary Ely are the same entity is 0.82 as Figure 6.3 shows. The two records in Figure 6.1 for Mary Ely *osmx132* and Mary Ely *osmx106* differ only in the *Child1Name* attribute, a low discriminating attribute when there is a mismatch because there may be many children of the same parents, all with different names. Figure 6.3 also shows that the second Mary Ely does not match with the first or third Mary Ely, both with a probability of 0.56. In Figure 6.1, the record for Mary Ely (*osmx202*) agrees with the other Mary Ely records only in its *Gender* field, which is not a discriminating field. Duke also concludes that the probability that the first and fourth Gerard Lathrop are the same is 0.82 as Figure 6.4 shows. In Figure 6.1 the first Gerard Lathrop (*osmx266*) and the fourth (*osmx296*) have identical records. The other Gerard Lathrop pairs differ in various ways.

Duke makes conclusions based on a threshold we set. For the results in Figures 6.3 and 6.4, a threshold of 0.80 lets Duke correctly conclude that the first and third Mary Ely are the same person, that the first and fourth Gerard Lathrop are the same person, and that all other persons differ from each other.

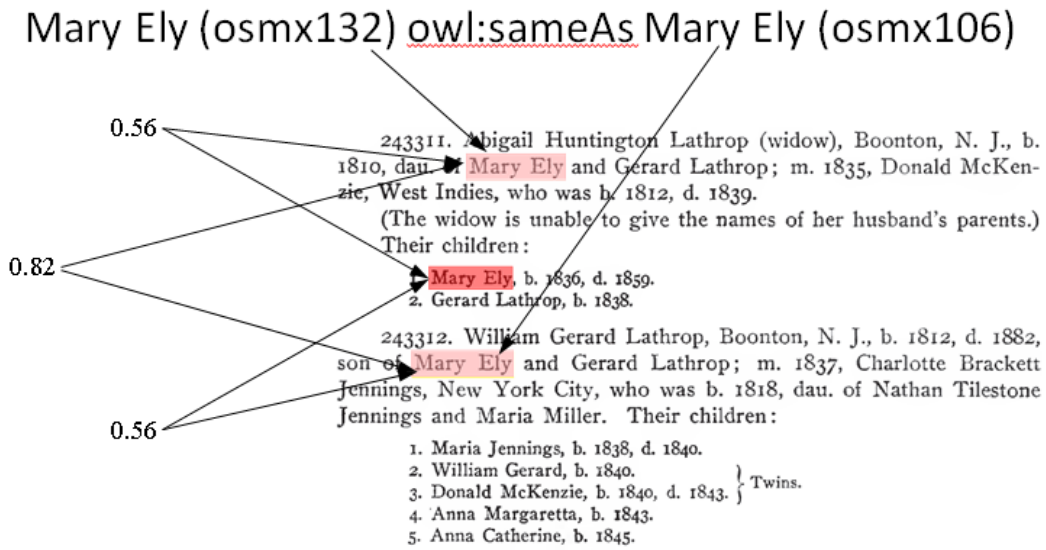

Figure 6.3: Match Probabilities for Mary Ely Resolution.

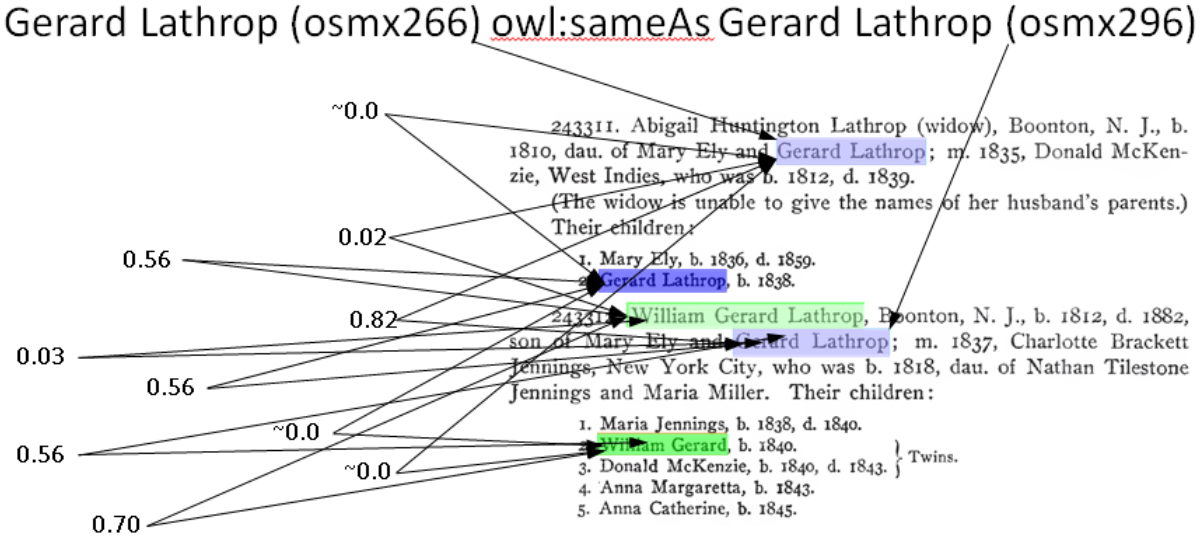

Figure 6.4: Match Probabilities for Gerard Lathrop Resolution.

Chapter 7

Case Studies

We present two case studies that document the process of extracting and organizing facts using FROntIER. The first case study is from *The Ely Ancestry* [BEV02], and the second is from the *Index to The Register of Marriages and Baptisms in the Parish of Kilbarchan, 1649–1772* [Gra12].

7.1 Case Study 1: *The Ely Ancestry*

Beginning with the part of text of the page in Figure 1.1 shown in Figure 7.1, we developed an initial full-line example of all three phases of FROntIER: extraction, inference, and identity resolution. Using the source and target conceptualizations in Figures 1.2 and 1.3, we tuned the extraction ontology's regular expressions, inference rules, and parameter settings to work well with the page excerpt in Figure 7.1.

To check the results, we developed an evaluation tool as part of the thesis. Figure 7.2 shows a screenshot of the metric calculator. It allows a user to select a populated gold-standard ontology to compare against a tool-generated populated ontology. In Figure 7.2 the gold-standard and FROntIER-generated ontologies are for a page in the *Kilbarchan Parish Record*. An existing Annotation Tool[1] designed specifically to create populated ontologies lets a user annotate a document page as a gold standard. The metric calculator also lets users choose which object and relationship sets to be evaluated (all of them have been selected

[1] dithers.cs.byu.edu/annotator2

243311. Abigail Huntington Lathrop (widow), Boonton, N. J., b. 1810, dau. of Mary Ely and Gerard Lathrop; m. 1835, Donald McKenzie, West Indies, who was b. 1812, d. 1839.

(The widow is unable to give the names of her husband's parents.)

Their children:

1. Mary Ely, b. 1836, d. 1859.
2. Gerard Lathrop, b. 1838.

243312. William Gerard Lathrop, Boonton, N. J., b. 1812, d. 1882, son of Mary Ely and Gerard Lathrop; m. 1837, Charlotte Brackett Jennings, New York City, who was b. 1818, dau. of Nathan Tilestone Jennings and Maria Miller. Their children:

1. Maria Jennings, b. 1838, d. 1840.
2. William Gerard, b. 1840. } Twins.
3. Donald McKenzie, b. 1840, d. 1843. }
4. Anna Margaretta, b. 1843.
5. Anna Catherine, b. 1845.

Figure 7.1: Excerpt from *The Ely Ancestry* Page 419.

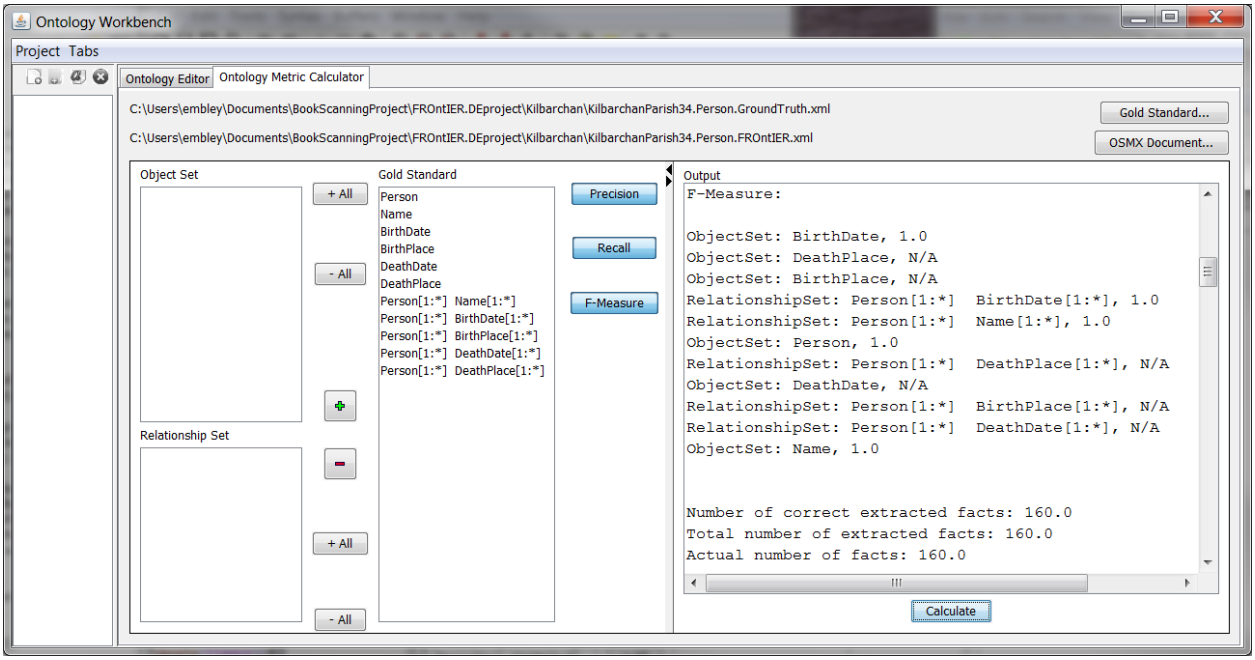

Figure 7.2: Screenshot of the Metric Calculator.

in Figure 7.2). Users may also select which evaluation metrics to use. Figure 7.2 shows an F-Measure result.

Extracted Facts	**Precision**	**Recall**	**F-measure**
Name	1.000	0.941	0.970
BirthDate	0.818	0.818	0.818
DeathDate	0.800	0.800	0.800
MarriageDate	0.500	0.500	0.500
Person-BirthDate	0.889	0.727	0.800
Person-DeathDate	0.750	0.600	0.667
Son-Person	1.000	1.000	1.000
Daughter-Person	1.000	1.000	1.000
Child-Person	1.000	0.857	0.923
Person-Spouse-MarriageDate	1.000	0.500	0.667
Inferred Facts	**Precision**	**Recall**	**F-measure**
Person-Name	0.958	0.902	0.929
Person-BirthDate	0.889	0.727	0.800
Person-DeathDate	0.750	0.600	0.667
Person-Gender	1.000	0.941	0.970
Person-Child	1.000	0.900	0.947
Person-Spouse-MarriageDate	1.000	0.900	0.947
Person_A same-as Person_B	1.000	1.000	1.000

Figure 7.3: Evaluation over the Excerpt in Figure 7.1.

Figure 7.3 shows the results of our initial development work applied to the page excerpt in Figure 7.1. We observe that the F-measures are near 100% for some of the extracted facts, but fall off for *MarriageDate* where we extracted only one of the two marriage dates and thus also fall off for *Person-Spouse-MarriageDate*. The recall result for *DeathDate* was lower than expected, having missed two of the five. Considering the F-measure for implied facts, note that added surnames for listed children in *Name*, added male and female designations in *Gender*, and added *same-as* predicates provide new information. The remainder are merely copies of extracted information with *Person-Child* being a copy from three to one (those extracted as sons and daughters as well as those extracted as children). From the F-measure results, we observe that inferred results mirror extracted results. This is true even for new information where the inferred results depend on correct extraction results.

Although it is possible for the encoding of rules or the setting of entity-resolution parameters to be incorrect, once they are debugged and properly tuned, there is essentially nothing more we can do to improve them. Thus, in the remainder of our case studies we focus on the extraction rules, where tuning them up for one page does not necessarily mean that they will have the same accuracy for other pages.

We next considered the whole Ely page in Figure 1.1 and produced the extraction ontology we have been considering as a running example. We then applied it blindly to two similar but randomly chosen Ely pages: Page 440 in Figure 7.4 and Page 479 in Figure 7.5. The results are respectively in Figure 7.6 and 7.7.

The F-Measures in Figure 7.6 are near 100% for several object sets and reasonably good for all object and relationship sets except *Daughter-Person*, where several OCR errors (e.g. "`<^au. of`", "`d^^^- of`", and "`<^au. of`") prevented the extraction-rule patterns from succeeding. Like the results for Page 440, the F-Measures for Page 479 in Figure 7.7 are almost all reasonably good. Several are 100% and even *Daughter-Person* is good at 0.818. However, the relationship set *Child-Person* has an F-Measure of only 0.200 due to a multitude of problems including OCR errors (particularly, "I." for "1." causing the first child in each list to be missed), a missing name of one of the children, and failure to extract one of the mother's names correctly.

SEVENTH GENERATION.

245891. Amelia Louisa Hovey, Brooklyn, N. Y., b. 1843, dau. of Henry Russell Hovey and Mary Emma Kutz; m. 1865, Roger Williams Love, who was b. 1842, d. 1879, son of Horace Thomas Love and Catherine Greene Waterman. Their children:

1. Henry Hovey, b. 1866.
2. Robert Harlow, b. 1868.
3. Winifred, b. 1870.
4. Roger, b. 1872.
5. Mabel, b. 1873.
6. Maude Marian, b. 1876.

245892. Mary Emma Hovey, Newark, N. J., who was b. 1845, dau. of Henry Russell Hovey and Mary Emma Kutz; m. 1861, James Thomas Curtin, London, England, who was b. 1831, d. 1862, son of James Curtin and Louisa Tequan. Their children:

1. James Hovey, b. 1865.

245894. Ada Louisa Hovey, New York City, b. 1849, dau. of Henry Russell Hovey and Mary Emma Kutz; m. 1870, George Metcalf Cone Richmond, 202 Wilson St., Brooklyn, E. D., N. Y., who was b. 1847, son of Nelson Clark Richmond and Mary Ann Collins Cone. Their children:

1. May Kate, b. 1871.

245895. Henry Russell Hovey, 211 Tremont St., Boston, Mass., b. 1851, son of Henry Russell Hovey and Mary Emma Kutz; m. 1880, Alice Eliza Huntley, Alstead, N. H., who was b. 1855, dau. of Elisha Alanson Huntley and Ruth Sleeper Gee. Their children:

1. Mary Emilie, b. Dec. 25, 1880.

246142. Mary Lord Ely, b. 1834, dau. of John Ely and Sarah Colt Lord; m. 1861, Charles Henry Payson, Rindge, N. H., who was b. 1831, d. 1877, son of Phillips Payson and Elizabeth Boutelle. Their children:

1. Charles Henry, b. 1864.
2. Sarah Lord, b. 1866.
3. Bessie Boutelle, b. 1869.
4. Edward Stiles Ely, b. 1871; d. 1880.
5. Mary Lord, b. 1874.

246143. John Andrews Ely, 22 Pine St., N. Y. City, b. 1836, son of John Ely and Sarah Colt Lord; m. 1876, Mary Elizabeth Berrien, who was b. 1843, dau. of Cornelius Anthony Berrien and Cornelia Clair Hartman. Their children:

1. John Andrews, b. 1878.
2. Sarah Berrien, b. 1879.

Figure 7.4: Page 440 of *The Ely Ancestry*

tor, N. Y., who was b. 1813, son of John Kinnan and Abigail Budd. Their children:

1. Hattie Olivia.
2. Carrie Emily.
3. Abby Budd.

1331514. Emma De Ette Doty, Clinton, Mich., b. 1842, dau. of Olivia Ely and George Doty; m. 1866, Benjamin James Bond, Clinton, Mich., b. 1838, son of Lewis Bond and Louisa Swartout. Their children:

1. Melinda Vance, b. 1868.
2. William Doty, b. 1871; d. 1872.
3. George Lewis.
4. Louis Oliva, b. 1876.

1331531. Fanny Ely, Hector, Schuyler Co., N. Y., b. 1842, dau. of Edward Hector Ely and Loretta Shannon; m. John Crisfield, Lodi, Seneca Co., N. Y.

1331532. William Ely, Hector, Schuyler Co., N. Y., b. 1843, son of Edward Hector Ely and Loretta Shannon; m. 1879, Delphina C. Bradner, Orange, N. Y., who was b. 1855. dau. of James Bradner and Clarissa Gatley. Their children:

1. ——— ———, b. 1879.

1331541. William Henry Smith, 171 So. Clinton St., Syracuse, N. Y., b. 1841, son of Caroline Ely and Edward Smith; m. 1869, Frances Emmeline Covell, Omaha, Neb., who was b. 1835, dau. of Lemuel Covell and Loranna Churchill. Their children:

1. Adelia Caroline, b. 1872.
2. Louis Covell, b. 1874.

1331542. Fanny Maria Smith, b. 1843, d. 1873, dau. of Caroline Ely and Edward Smith; m. 1865, William Edward Keeler, Moravia, N. Y., who was b. 1842, son of Thompson Keeler and Eliza Allee. Their children:

1. Ralph Richard, b. 1866.

1331543. Richard Smith, Kewaunee, Wis., b. 1845, son of Caroline Ely and Edward Smith; m. 1877, Martina Willis Read, New York City, who was b. 1849, dau. of Martin Willis Read and Catherine Divens. No children.

Figure 7.5: Page 479 of *The Ely Ancestry*

Extracted Facts	Precision	Recall	F-Measure
Name	1.000	1.000	1.000
BirthDate	1.000	0.964	0.982
DeathDate	1.000	1.000	1.000
MarriageDate	1.000	1.000	1.000
Child	1.000	0.813	0.897
Person-BirthDate	0.920	0.821	0.868
Person-DeathDate	0.750	0.750	0.750
Son-Person	1.000	0.833	0.909
Daughter-Person	0.667	0.333	0.444
Child-Person	0.792	0.594	0.679
Person-Spouse-MarriageDate	1.000	0.500	0.667

Figure 7.6: Evaluation over page 440 of *The Ely Ancestry*

Extracted Facts	Precision	Recall	F-Measure
Name	0.957	0.957	0.957
BirthDate	1.000	1.000	1.000
DeathDate	1.000	1.000	1.000
MarriageDate	1.000	1.000	1.000
Child	0.600	0.750	0.667
Person-BirthDate	0.769	0.556	0.645
Person-DeathDate	1.000	1.000	1.000
Son-Person	1.000	1.000	1.000
Daughter-Person	0.900	0.750	0.818
Child-Person	0.500	0.125	0.200
Person-Spouse-MarriageDate	1.000	0.800	0.889

Figure 7.7: Evaluation over page 479 of *The Ely Ancestry*

7.2 Case Study 2: *Kilbarchan Parish Record*

In our second case study, we created three extraction ontologies: (1) persons with their vital information (Figure 7.8), (2) couples with marriage date and place (Figure 7.10), and (3) parents with children in families (Figure 7.9). We tuned the regular-expression extraction rules using Page 31 (Figure 7.11) and applied it blindly to Pages 32 (Figure 7.12) and 96 (Figure 7.13). Figures 7.14–7.22 show the results, which are good, many being near 100%.

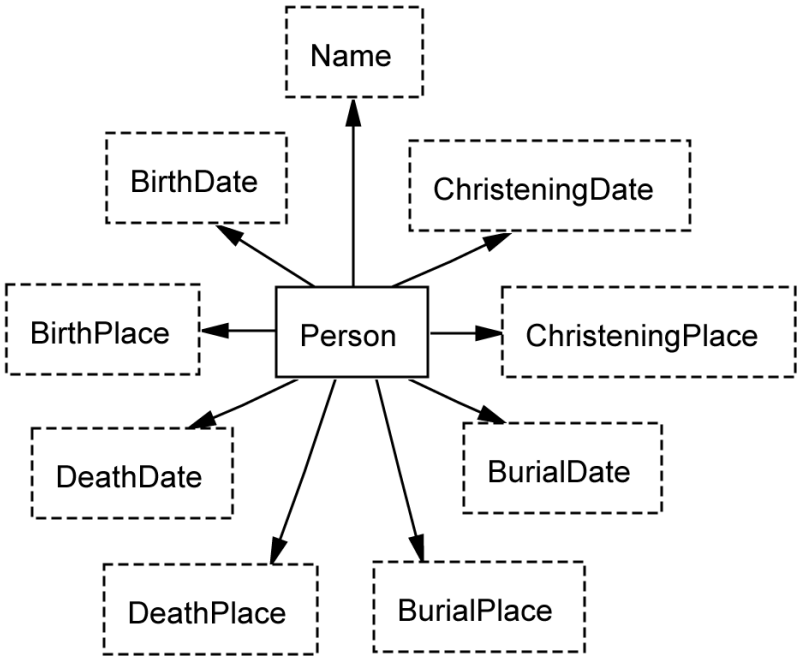

Figure 7.8: Extraction Ontology for Persons and their Vital Information.

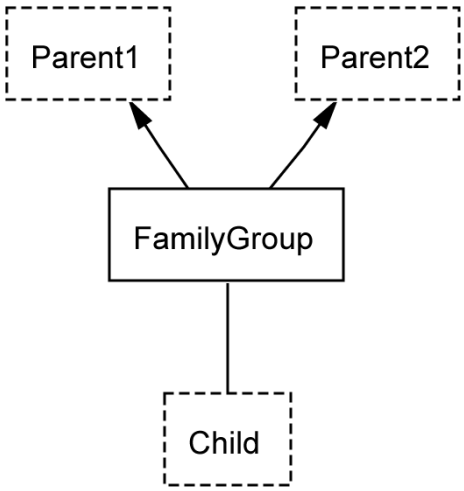

Figure 7.9: Extraction Ontology for Families.

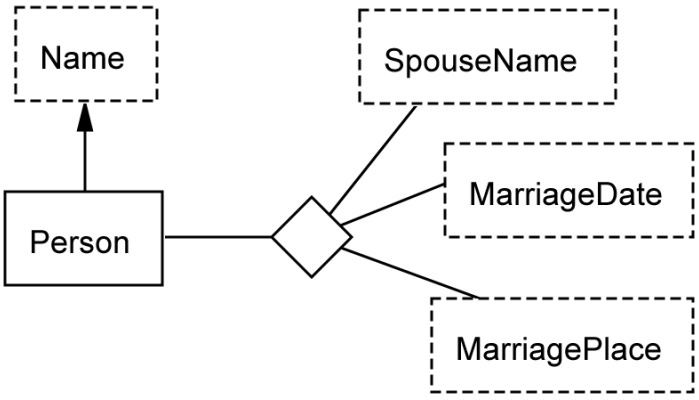

Figure 7.10: Extraction Ontology for Marriages.

Jean, 6 Mar. 1698.
Ann, 25 Oct. 1701.
Cordoner, James, par., and Florence Landiss, par. of Paisley
m. 13 June 1679
Cordoner, John, and Catherine Adam m. 21 April 1656
Cordonnar, John, par., and Jean Craufurd, par. of Beith
m. Beith, 16 June 1659
Cordoner, John, and Issobell Speir, in Walkmilne of Johnstoun
m. 16 July 1673
Jean, 17 May 1674.
James, 6 Oct. 1676.
Agnes, 24 Jan. 1679.
Jonat, 24 June 1681.
John, 15 July 1683.
Cordoner, John, and Margaret Cochran, in Nether Walkmilne
William, 13 Mar. 1681.
Cordner, John, and Jonet Cochran, in Walkinshaw, 1688 in
Walkmiln of Johnstoun m. 22 April 1680
Jonet, 3 Dec. 1682.
Thomas, 7 Aug. 1688.
Margaret, 16 Dec. 1692.
Jean, 16 Feb. 1696.
Cordonar, William, and Jean Cochran m. 7 Feb. 1651
Cordoner, William
William, 1 Aug. 1651.
William, 2 Jan. 1653.
Janet, 26 Feb. 1654.
Cordoner, William, in Achindinane
John, 10 Nov. 1654.
Cordonar, William, in Over Wakmilne of Johnstoun
Margaret, 27 July 1655.
Jean, 25 Sept. 1657.
Margaret, 24 June 1660.
Cordonner, William, in Achinames.
Jane, 23 April 1658.
James, 29 May 1659.
Cordownar, William, in Nether Walkmylne
Thomas, 25 Sept. 1657.
Cordoner, William, and Issobell Young, in Auchnames
Jean, 2 Oct. 1674.
Cordoner, William, at the Wakmilns of Johnstoun
Margaret, 9 Dec. 1688.
Cordoner, William, and Eliza Orr, in Netherwalkmilne of Johnstoun
Agnes, 15 Feb. 1691.
Jean, 10 Feb. 1693.
Eliza, 28 July 1695.
Jean, 31 July 1698.
Corss, John, and Jean Patison
Jonet, 28 July 1682.
Couper, James, and Issobell Load m. 30 Nov. 1682
Couper, James, par. of Erskine, and Mary Black, par. 30 Mar. 1744
Coupar, William, in Kilbarchan, and Janet Caldwell p. 29 Dec. 1768
John, 6 Nov. 1769.
Cowan, Daniel, in town par. of Paisley, and Margaret Dougal
p. 15 Jan. 1763
Craig, James, par. of Kilbryde, and Jonet Cordonar, par.
m. 28 June 1658
Craig, James, Moreland in Forehouse, and Jonet Reid
m. Lochwinnoch, 18 Jan. 1693
James, 31 May 1695.
Margaret, 19 Sept. 1697.

Figure 7.11: Page 31 of the *Kilbarchan Parish Record*.

Craig, James, and Mary Barr
John, 30 May 1743.
Craig, James, and Elizabeth Story, 1751 in Law
Elizabeth, 14 Aug. 1748.
Margaret, 3 Feb. 1751.
Robert, born 29 July 1753.
John, 25 Jan. 1756.
Craig, James, and Mary M'Dowall, in Monkland p. 8 Dec. 1749
Janet, born 12 July 1751.
James, 8 April 1757.
Craig, John, par. of Beith, and Marione Speir m. 18 Dec. 1672
Craig, John, and Janet Reid, in Forehouse
Mary, 20 Oct. 1673.
Craig, John, and Isobell Merchant m. 15 June 1682
Craig, John, and Elizabeth Kirk, who came from Ireland
Elizabeth, 12 Oct. 1690.
Craig, John, and Marion Clark, in Sweinlees, par. of Paisley
Samuel, 14 June 1691.
Craig, John, par. of Neilstoun, in Cartside, and Margaret King
m. 8 Feb. 1694
Robert, 6 Dec. 1694.
Mary, 4 Dec. 1698.
Craig, John, and Margaret Robison
Katherine, 18 Jan. 1741.
Craig, John, par., and Elizabeth Storie, in Abbey par. of Paisley
p. 30 May 1747
Craig, Thomas, in Kilbarchan, and Elizabeth M'Caslane
Agnes, born 8 July 1759.
Craig, Thomas, in Kilbarchan, and Janet Crawford p. 29 May 1762
Thomas, born 8 Jan. 1764.
Craig, William, and Agnes Duff m. 25 May 1654
Craig, William, in Kirktoune
William, 30 Sept. 1655.
Jean, 25 July 1658.
Craig, William, and Marion Broune, in Locherside
Marion, 14 May 1676.
Craig, William, and Margaret Dick, in Kirkton, 1692 in
Locherside, 1695 par. of Houstoun m. 29 April 1681
William, 5 Feb. 1682.
Elizabeth, 2 Sept. 1692.
Janet, 28 April 1695.
Craig, William, and Agnes Park, in Milne of Johnstoun
m. 12 Nov. 1689
Jean, 28 Dec. 1690.
William, 4 Mar. 1692.
James, 28 Oct. 1694.
Mary, 18 April 1697.
William, 5 Jan. 1701.
Craig, William, in Braes, and Janet Kerr
Jane, born 18 Dec. 1757.
James, born 6 May 1760.
Craig, William, in Halhill, and Janet Inglis
Jane, born 20 Nov. 1763.
Craig, William, and Anne Lang p. 7 June 1771
Crawford, Alexander, and Janet Whithill p. 18 July 1772
Crawford, Duncan, and Mary Neil p. 6 April 1753
Crawford, John in Houstoun
Marion, 18 Feb. 1653.
Daniel, 9 Feb. 1655.
Crauford, John, par. of Beith, and Anna Lyle, par.
m. Kilellan, 31 July 1683

Figure 7.12: Page 32 of the *Kilbarchan Parish Record*.

Rose, Robert, in Linwood
Elizabeth, 2 Nov. 1655.
James, 22 Aug. 1658.
Rosse, Robert, in Meikle Fultoune
Robert, 12 May 1661.
Ross, Robert, of Kirkland, and Katherine Hamilton, 1688 in Linwood m. 26 April 1677
Grissell, 31 Dec. 1682.
Elizabeth, 9 Dec. 1688.
Agnes, 4 Dec. 1692.
Ross, Robert, and Mary Colquhoun, in Linwood
Christian, 4 June 1697.
Russide, David, in Hill, and Margaret Stuart
Anne, born 7 Sept. 1767.
Sandilands, Thomas, surgeon in Kilbarchan, and Janet Lewis
Margaret, born 9 Feb. 1751.
John, born 8 Mar. 1753.
Sclatter, John, and Jonet Cochran, in Lochwinnoch
Margaret, 4 Feb. 1683.
Sklaitter, Peter, 1655 in Linwood
John, 3 July 1653.
Jonet, 30 May 1655.
Robert, 30 July 1658.
David, 7 April 1661.
Scott, ——, in Plainlees, and Margaret Barbour
——, 31 July 1709.
Scott, Alexander, and Agnes Greive, in Kilmacolm
James, —— June 1683
Scott, Archibald, par. of Largs, and Elizabeth Houstoun, par. in Kirktoun, 1695 in Craigends, 1698 in Kirktoun m. 31 Dec. 1691
Archibald, 24 Feb. 1693.
Francis, 4 Aug. 1695.
Catherine and Anna, 15 Nov. 1698.
John, 3 Jan. 1701.
Scott, John, in Kilbarchan, and Isabella Cumming, par. of Lochwinnoch p. 29 July 1749
Archibald, 17 June 1750.
Janet, born 23 Oct. 1752.
Scott, John, and Janet Wilson p. 19 April 1766
Semple or Sempill, Andrew, and Margaret Waterstoune m. 24 Feb. 1654
Semple, Andrew, in Craigens
Jane, 31 Dec. 1655.
Semple, Andrew, at Mill of Cart
James, 28 Dec. 1656.
Semple, Andro, in Erskinefauld
Elizabeth, 21 April 1661.
Semple, Francis, yr., of Beltrees, and Jean Campbell, par. of Lochgilphead m. Lochgilphead 3 April 1655
Robert, 11 April 1656.
James, 10 May 1657.
Semple, Hugh, in the Kirktoune
Thomas, 2 Nov. 1651.
Robert, 16 July 1653.
Hew, 18 Mar. 1655.
Semple, Hew, in Boghouse
William, 3 April 1657.
Semple, Hugh, of Waterstoune
James, 8 Mar. 1660.
Marie, 20 July 1662.
Sempill, Hugh, in Mossend, and Elizabeth Hall
Hugh, 17 June 1705.

Figure 7.13: Page 96 of the *Kilbarchan Parish Record*.

Extracted Facts	Precision	Recall	F-Measure
Name	1.000	1.000	1.000
BirthDate	N/A	N/A	N/A
BirthPlace	N/A	N/A	N/A
DeathDate	N/A	N/A	N/A
DeathPlace	N/A	N/A	N/A
ChristeningDate	1.000	1.000	1.000
ChristeningPlace	N/A	N/A	N/A
BurialDate	N/A	N/A	N/A
BurialPlace	N/A	N/A	N/A
Person-BirthDate	N/A	N/A	N/A
Person-BirthPlace	N/A	N/A	N/A
Person-DeathDate	N/A	N/A	N/A
Person-DeathPlace	N/A	N/A	N/A
Person-ChristeningDate	1.000	1.000	1.000
Person-ChristeningPlace	N/A	N/A	N/A
Person-BurialDate	N/A	N/A	N/A
Person-BurialPlace	N/A	N/A	N/A

Figure 7.14: Person Extraction Results from Page 31 of the *Kilbarchan Parish Record*.

Extracted Facts	Precision	Recall	F-Measure
Name	1.000	1.000	1.000
BirthDate	1.000	1.000	1.000
BirthPlace	N/A	N/A	N/A
DeathDate	N/A	N/A	N/A
DeathPlace	N/A	N/A	N/A
ChristeningDate	1.000	1.000	1.000
ChristeningPlace	N/A	N/A	N/A
BurialDate	N/A	N/A	N/A
BurialPlace	N/A	N/A	N/A
Person-BirthDate	1.000	1.000	1.000
Person-BirthPlace	N/A	N/A	N/A
Person-DeathDate	N/A	N/A	N/A
Person-DeathPlace	N/A	N/A	N/A
Person-ChristeningDate	1.000	1.000	1.000
Person-ChristeningPlace	N/A	N/A	N/A
Person-BurialDate	N/A	N/A	N/A
Person-BurialPlace	N/A	N/A	N/A

Figure 7.15: Person Extraction Results from Page 32 of the *Kilbarchan Parish Record*.

Extracted Facts	Precision	Recall	F-Measure
Name	1.000	0.943	0.971
BirthDate	1.000	1.000	1.000
BirthPlace	N/A	N/A	N/A
DeathDate	N/A	N/A	N/A
DeathPlace	N/A	N/A	N/A
ChristeningDate	1.000	0.935	0.967
ChristeningPlace	N/A	N/A	N/A
BurialDate	N/A	N/A	N/A
BurialPlace	N/A	N/A	N/A
Person-BirthDate	1.000	1.000	1.000
Person-BirthPlace	N/A	N/A	N/A
Person-DeathDate	N/A	N/A	N/A
Person-DeathPlace	N/A	N/A	N/A
Person-ChristeningDate	1.000	0.906	0.951
Person-ChristeningPlace	N/A	N/A	N/A
Person-BurialDate	N/A	N/A	N/A
Person-BurialPlace	N/A	N/A	N/A

Figure 7.16: Person Extraction Results from Page 96 of the *Kilbarchan Parish Record*.

Extracted Facts	Precision	Recall	F-Measure
Name	1.000	0.938	0.968
SpouseName	1.000	0.938	0.968
MarriageDate	1.000	0.917	0.957
MarriagePlace	1.000	1.000	1.000
Person-SpouseName-MarriageDate-MarriagePlace	1.000	0.938	0.968

Figure 7.17: Marriages Extraction Results from Page 31 of the *Kilbarchan Parish Record*.

Extracted Facts	Precision	Recall	F-Measure
Name	1.000	1.000	1.000
SpouseName	1.000	1.000	1.000
MarriageDate	1.000	0.917	0.957
MarriagePlace	N/A	N/A	N/A
Person-SpouseName-MarriageDate-MarriagePlace	0.972	0.972	0.972

Figure 7.18: Marriages Extraction Results from Page 32 of the *Kilbarchan Parish Record*.

Extracted Facts	Precision	Recall	F-Measure
Name	1.000	0.846	0.917
SpouseName	1.000	0.846	0.917
MarriageDate	1.000	0.500	0.667
MarriagePlace	0.000	0.000	0.000
Person-SpouseName-MarriageDate-MarriagePlace	0.924	0.782	0.847

Figure 7.19: Marriages Extraction Results from Page 96 of the *Kilbarchan Parish Record*.

Extracted Facts	Precision	Recall	F-Measure
Parent1	0.923	0.857	0.889
Parent2	0.714	0.556	0.625
Child	1.000	0.867	0.929
FamilyGroup-Parent1	0.923	0.857	0.889
FamilyGroup-Parent2	0.714	0.556	0.625
FamilyGroup-Child	1.000	0.867	0.929

Figure 7.20: Family Extraction Results from Page 31 of the *Kilbarchan Parish Record*.

Extracted Facts	Precision	Recall	F-Measure
Parent1	0.833	0.588	0.690
Parent2	0.800	0.533	0.640
Child	1.000	0.690	0.816
FamilyGroup-Parent1	0.833	0.588	0.690
FamilyGroup-Parent2	0.800	0.533	0.640
FamilyGroup-Child	1.000	0.690	0.816

Figure 7.21: Family Extraction Results from Page 32 of the *Kilbarchan Parish Record*.

Extracted Facts	Precision	Recall	F-Measure
Parent1	1.000	0.737	0.848
Parent2	1.000	0.455	0.625
Child	1.000	0.657	0.793
FamilyGroup-Parent1	1.000	0.737	0.848
FamilyGroup-Parent2	1.000	0.455	0.625
FamilyGroup-Child	1.000	0.657	0.793

Figure 7.22: Family Extraction Results from Page 96 of the *Kilbarchan Parish Record*.

Chapter 8

Conclusions and Future Work

FROntIER accomplishes the task of automatically extracting stated facts, inferring implicit facts, and resolving object references in a synergistic framework. Results for the thesis include the following:

1. We have created recognizers for non-lexical object sets, relationship sets, and ontology snippets, and we have added them to the existing lexical-object-set recognizers in OntoES [ECJ+99]. Thus, extraction ontologies have been significantly augmented to allow for more complicated relationships to be extracted.
2. We have added provisions for inferring and organizing facts through rules and the Jena reasoner, which allows for implicit facts to be obtained from stated facts and the reorganization of facts from a source extraction ontology to a target ontology.
3. We have integrated Duke into FROntIER to resolve object identity.
4. We conducted several case studies to exercise the features of FROntIER.

The results of the case studies show promise. FROntIER can extract and organize facts to a reasonable degree of accuracy, which allows for a greater use of the facts and implied facts found in OCRed historical documents.

In Chapter 7 we gave an explanation of running FROntIER over various pages from *The Ely Ancestry* and the *Kilbarchan Parish Record* and how the results of evaluation were obtained. What was not explained are the details of how the recognizers were produced and how improvements were made to the recognizers at each step. It was found to be most

effective to begin writing recognizers for the lexical object sets followed by the non-lexical object sets and then proceed to recognizers for relationship sets. Recognizers for ontology snippets were only effective for certain types of records—list-like records like the child lists in Figure 1.1.

We found the task of automatically extracting stated facts, inferring implicit facts, and resolving object references to be difficult. Each step is dependent on the previous step, and the accuracy of extraction or organization at each step affects the results of the next step. In order to achieve results similar to manual annotation, much more work will need to be done. One way to ease the burden of extraction-rule creation would be to add provisions for more generalized ontology snippet recognizer patterns, which would allow general extraction of records in nested lists like the nested child lists in Figure 1.1. Currently a separate pattern must be written to connect each child in the nested list to the parents in the list header. More generally, a machine-learning component could be added to FROntIER that would recognize patterns in text to ease the burden of manual entry of recognizers.

Much more work on the areas that FROntIER covers was done by us than is stated in the body of this thesis. We made attempts at extracting and organizing facts from a subset of the LDS Church's online repository (which has 90,000+ documents). Evaluation results were obtained by annotating the documents using an annotation tool created by the Data Extraction Research Group at BYU and evaluating against the results produced by FROntIER. The annotations were produced by students of one of the professors in the research group, and the forms used for the annotations were designed solely with hand-annotation in mind, which then required inference rules to be written for transforming the extracted facts to fit into the target ontology in Figure 1.2. There were many issues along the way, and the task was much too large for the scope of this thesis. It did, however, provide a window into the problems that remain to be solved in order to produce human-like accuracy of extracting and organizing facts in real-world historical documents. Some of these problems include the lack of author grammar consistency, OCR errors, inconsistent text

layout, abbreviations, missing information, natural language, and use of pronouns instead of names of entities. Still, given the complexity of this task, FROntIER provides an initial automated way to extract and organize the facts found in the Church's online repository of books.

References

[AM98] G.O. Arocena and A.O. Mendelzon. WebOQL: Restructuring documents, databases, and webs. In *Proceedings of the 14th IEEE International Conference on Data Engineering*, pages 24–33, Orlando, Florida, 1998.

[BBC+10] L. Blanco, M. Bronzi, V. Crescenzi, P. Merialdo, and P. Papotti. Redundancy-driven web data extraction and integration. In *Proceedings of the 13th International Workshop on the Web and Databases*, pages 1–6, New York, NY, USA, June 2010.

[BCM+03] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

[BEV02] M.S. Beach, W. Ely, and G.B. Vanderpoel. *The Ely Ancestry*. The Calumet Press, 1902.

[BGH09] R. Baumgartner, G. Gottlob, and M. Herzog. Scalable web data extraction for online market intelligence. *Proceedings of the VLDB Endowment*, 2(2):1512–1523, 2009.

[BHH+11] D. Burdick, M. Hernández, H. Ho, G. Koutrika, R. Krishnamurthy, L. Popa, I.R. Stanoi, S. Vaithyanathan, and S. Das. Extracting, linking and integrating data from public sources: A financial case study. *IEEE Data Engineering Bulletin*, 34(3):60–67, 2011.

[Chr12] P. Christen. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer, 2012.

[CKGS06] C-H. Chang, M. Kayed, M.R. Girgis, and K. Shaalan. Survey of web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1411–1428, 2006.

[CM98] V. Crescenzi and G. Mecca. Grammars have exceptions. *Information Systems*, 23(8):539–565, 1998.

[ECJ+99] D.W. Embley, D.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 pages. *Data & Knowledge Engineering*, 31(3):227–251, 1999.

[EKW92] D.W. Embley, B.D. Kurtz, and S.N. Woodfield. *Object-oriented Systems Analysis: A Model-Driven Approach*. Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[ELL11] D.W. Embley, S.W. Liddle, and D.W. Lonsdale. Conceptual modeling foundations for a web of knowledge. In D.W. Embley and B. Thalheim, editors, *Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges*, chapter 15, pages 477–516. Springer, 2011.

[Emb80] D.W. Embley. Programming with data frames for everyday data items. In *Proceedings of the AFIPS National Computer Conference*, pages 301–305, Anaheim, CA, USA, May 1980.

[EZ10] D.W. Embley and A. Zitzelberger. Theoretical foundations for enabling a web of knowledge. In *Proceedings of Foundations of Information and Knowledge Systems*, pages 211–229, Sofia, Bulgaria, February 2010.

[Gra12] R.J. Grant. *Index to The Register of Marriages and Baptisms in the Parish of Kilbarchan, 1649–1772*. J. Skinner & Company, LTD., 1912.

[GX09] J. Gardner and L. Xiong. An integrated framework for de-identifying unstructured medical data. *Data & Knowledge Engineering*, 68(12):1441–1451, 2009.

[HMGM97] J. Hammer, J. McHugh, and H. Garcia-Molina. Semistructured data: The TSIMMIS experience. In *Proceedings of the 1st East-European Symposium on Advances in Databases and Information Systems*, pages 1–8, St. Petersburg, Russia, 1997.

[HSW07] T.N. Herzog, F.J. Scheuren, and W.E. Winkler. *Data Quality and Record Linkage Techniques*. Springer, 2007.

[LHE03] S.W. Liddle, K.A. Hewett, and D.W. Embley. An integrated ontology development environment for data extraction. In *Proceedings of 2nd International Conference on Information Systems Technology and its Applications*, pages 21–33, Kharkiv, Ukraine, June 2003.

[LPH00] L. Liu, C. Pu, and W. Han. XWRAP: An XML-enabled wrapper construction system for web information sources. In *Proceedings of the 16th IEEE International Conference on Data Engineering*, pages 611–621, San Diego, California, 2000.

[MK11] R.B. Mishra and S. Kumar. Semantic web reasoners and languages. *Artificial Intelligence Review*, 35(4):339–368, 2011.

[SA01] A. Saiiuguet and F. Azavant. Building intelligent web applications using lightweight wrappers. *Data and Knowledge Engineering*, 36(3):283–316, 2001.

[Sar08] S. Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3):261–377, 2008.

[Sch12] P. Schone. Personal communication, 2012.

[TAC06] J. Turmo, A. Ageno, and N. Català. Adaptive information extraction. *ACM Computing Surveys*, 38(2):1–47, 2006.

[WLE05] A. Wessman, S.W. Liddle, and D.W. Embley. A generalized framework for an ontology-based data-extraction system. In *Proceedings of 4th International Conference on Information Systems Technology and its Applications*, pages 239–253, Palmerston North, New Zealand, May 2005.

[ZELS14] A.J. Zitzelberger, D.W. Embley, S.W. Liddle, and D.T. Scott. HyKSS: Hybrid keyword and semantic search. *Journal on Data Semantics*, 2014. (in press).