Enabling Data Quality with Lightweight Ontologies

Clint Bidlack

ActivePrime Inc.
Pasadena, CA
bidlack@activeprime.com

Abstract

As the volume and interconnectedness of corporate data grows, data quality is becoming a business competency essential to success. Existing methods for managing data quality do not scale up to large volumes of data in a way that is directly manageable by the owner of the data. For the past two years a new breed of data quality products, built on applied AI techniques, are empowering non-technical users. Over 150 businesses are benefiting from these products including NASDAQ, Visa, Experian, Oracle, Fidelity, Bank of America, Volvo, Dell, Sabic, and Dassault Systems. The applied AI techniques described include lightweight ontologies to efficiently find inexact textual matches in large data sets.

Introduction

Several business and technology drivers are disrupting the world of enterprise software and IT. These drivers include business acceptance of software-as-a-service (SaaS), wide adoption of the web as a platform, collapse of enterprise application silos, aggregation of data from disparate internal and external sources, the agile mindset, and the business conditions driving this agility.

These trends are resulting in more effective use of enterprise software as well as more efficient business operations. Effectiveness is driving adoption across the business landscape, across industries, and from very small companies up to the global 2000. Efficiency is driving application acceptance and usage within the company. This combination of effectiveness and efficiency, driving adoption and usage, is fueling enormous growth of structured business data.

All this structured business data requires maintenance, and that includes maintaining the quality of the data. For instance, with SaaS based CRM systems, ActivePrime and its partners have found that data quality has become the number one issue that limits return on investment, and as the volume of data grows, the pain experienced from poor quality data grows more acute. Data quality has been an ongoing issue in the IT industry for the past 30 years, and it continues and is expanding as an issue, fueling the growth of the data quality industry to $1B in 2008. It is also estimated that companies are losing 6% of sales because of poor management of customer data (Experian QAS 2006).

Copyright © 2009, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Existing Solutions

Old solutions to the problem were driven in part by the economics of the institutions having the problem. Very large organizations made up the bulk of the need for higher data quality. For the most part this was limited to large, Global 2000 corporations. They had the resources to deploy complex software system for gathering data, and they had the resources to solve the inevitable data quality problems resulting from this complexity. The traditional solutions relied heavily on manual operations in two different respects.

First, data was often hand-cleansed by contracting with external staffing agencies that had access to a large supply of lower cost labor. Business analysts would first identify what type of data quality work needed to be performed and on which data. Large data sets would be broken up into reasonable sizes and put into spreadsheets. This data would be distributed to the individual workers along with instructions for cleansing. After the manual work on a spreadsheet was finished, oftentimes the work could be cross-checked by another worker and any discrepancies investigated. Once the data was finished it was reassembled into the appropriate format for loading back into the source IT system. Such manual effort has clear drawbacks, including the time required to cycle through the entire process, the possibility for manual error, and the need to export and then import the final results. Exporting is usually fairly easy. The importing is almost always the bigger issue. Import logic typically needs to identify not only which specific fields and records to update, but also how to deal with deleted or merged data, and that all needs to happen error free. This result in additional work required to build and thoroughly test the import tools. Finally, the entire manual process has little room for increased return on investment. The customer has to pay for the manual work each time data is cleansed meaning that the attractive economic benefits of automation are not realized.

Second, earlier data quality vendors like FirstLogic, Trillium, GroupOne, and others provided technological solutions to data quality problems, and these required significant manual setup effort. The reasons for the manual setup included business analysis to understand data quality needs of the organization, identification of the final data quality work flow, and then the actual programming and configuration to put the data quality solution in place. In other words, these companies were building custom data quality solutions using data quality vendor APIs. Once the solution was put in place, automation reduced the manual effort, so a longer horizon for return on investment was acceptable. This worked fine for large companies that could afford both the problem, initial enterprise system that aggregates the data, and the solutions. Today, sophisticated business applications are being used by even the smallest organizations; hence, the manual effort associated with data quality must be in alignment with the resources of these smaller organizations. The data quality solutions must leverage automation, and at the same time provide the business user with intuitive access to the processing results and the ability to over ride the results.

ActivePrime Solutions

ActivePrime's initial product and services focus on increasing the quality of data in CRM systems. CRM systems store people-centric data at the core with other types of data, like product, transactional, or other forms of data considered secondary in nature. Though the secondary data is essential as well, in a CRM system the user interactions predominantly revolve around managing and querying the people-centric data. The main entities involved are organizations and people. Organizations are typically referred to as accounts and people as contacts. As is expected, accounts and contacts have properties such as postal addresses, email addresses, phone numbers and such. ActivePrime products focus on the effective and efficient management of the quality of account and contact data.

Before describing the various ActivePrime solutions, it is important to understand why the quality of data in CRM systems is so poor in the first place, and also how poor data enters the system. One of the main reasons for poor quality data in CRM systems is that the data is initially hand entered into some system, and often times the user is not aware of why the quality of the data is so important. The user may be initially entering the data into an email client where they are the only consumer of the data. Hence the quality of the data is not so important. If the company name is misspelled, or if the person's title is abbreviated in a unique way, there really is no problem. The issues arise if that data is then synchronized with a large volume of data in a CRM system with many other users, and the organization is depending on the data for operational purposes. Suddenly the misspellings, abbreviations, and missing information have significant material impact on the ability of the organization to leverage the data.

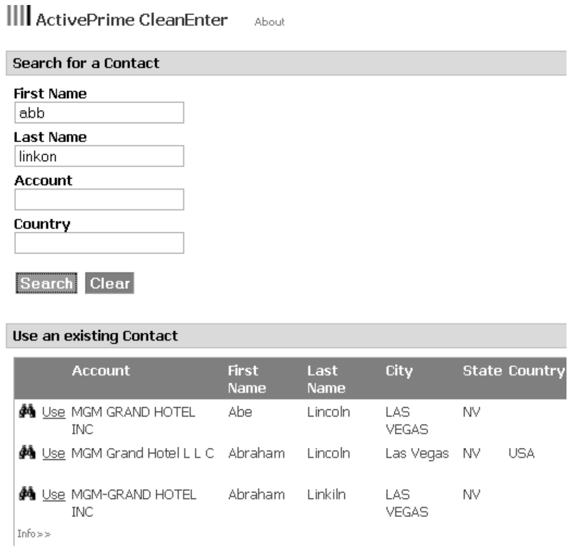
There are five points of entry for the majority of data entering CRM systems:

1. Manual entry of data by sales, marketing and support personnel.
2. Batch importing of data from external sources of information.
3. Data synchronization from other software and devices like email clients, mobile phones, and PDAs.
4. Web entry of data by a potential customer or other third party.
5. Migration of data from legacy CRM systems.

Successful CRM data quality initiatives must account for managing the quality of data at all these points of entry. ActivePrime's three solutions, CleanEnter, CleanCRM, and CleanMove, address each of these areas.

Data Quality Upon Manual Entry

Upon manual entry of data, either via the web or using the CRM data entry user interface, ActivePrime CleanEnter provides search functionality that notifies the user if they are about to enter a duplicate record. Figure 1 displays the search UI on contact entities and shows that entering a new contact of "Abb Linkon" has a potential match with three records in the system. At this point the user can drill down on a record, navigate to a record in the CRM system, or chose to continue entering "Abb Linkon" as a new record.



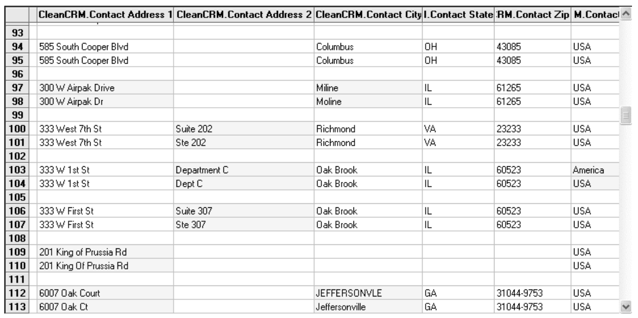
First Name: abb
Last Name: linkon
Account:
Country:

	Account	First Name	Last Name	City	State	Country
Use	MGM GRAND HOTEL INC	Abe	Lincoln	LAS VEGAS	NV	
Use	MGM Grand Hotel L L C	Abraham	Lincoln	Las Vegas	NV	USA
Use	MGM-GRAND HOTEL INC	Abraham	Linkiln	LAS VEGAS	NV	

Figure 1: Results of real time inexact matching.

Data Quality Upon Batch Entry

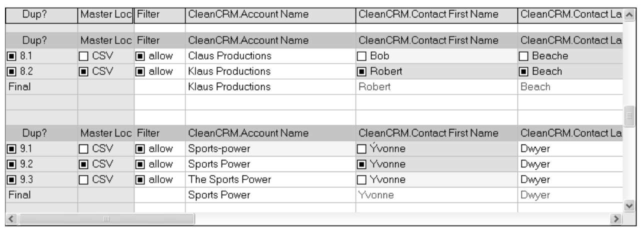
Customer data is often batch entered after going to a trade show or after purchasing a list of customer data from a third party. Upon batch entry of data, or when cleansing large volumes of data that have just been synchronized into the CRM system, sales or marketing support staff will need to identify areas in the data that need to be standardized or find and merge duplicate data. Figure 2 shows the standardization screen in ActivePrime CleanCRM with each set of two records being the record before standardization and then after. Rows 97 and 98 have two fixes, the address convention is standardized to short form, Dr instead of Drive, and the city name was fixed from Miline to Moline. Rows 100 through 113 show several other examples of address standardization including normalizing America to USA in 103 and 104. The user has control over which standardizations can be applied by selecting controls on a UI (not shown) including all standardizations being on or off, setting naming conventions for corporate names, business designations, address conventions, secondary address formatting, city spelling correction, and state conventions as short form or long form.



	CleanCRM.Contact Address 1	CleanCRM.Contact Address 2	CleanCRM.Contact City	l.Contact State	RM.Contact Zip	M.Contact
93						
94	585 South Cooper Blvd		Columbus	OH	43085	USA
95	585 South Cooper Blvd		Columbus	OH	43085	USA
96						
97	300 W Airpak Drive		Miline	IL	61265	USA
98	300 W Airpak Dr		Moline	IL	61265	USA
99						
100	333 West 7th St	Suite 202	Richmond	VA	23233	USA
101	333 West 7th St	Ste 202	Richmond	VA	23233	USA
102						
103	333 W 1st St	Department C	Oak Brook	IL	60523	America
104	333 W 1st St	Dept C	Oak Brook	IL	60523	USA
105						
106	333 W First St	Suite 307	Oak Brook	IL	60523	USA
107	333 W First St	Ste 307	Oak Brook	IL	60523	USA
108						
109	201 King of Prussia Rd					USA
110	201 King Of Prussia Rd					USA
111						
112	6007 Oak Court		JEFFERSONVLE	GA	31044-9753	USA
113	6007 Oak Ct		Jeffersonville	GA	31044-9753	USA

Figure 2: Standardization of address data during batch processing.

The second type of cleansing applied by CleanEnter is identification and merging of duplicate data. Duplicates can be identified based on exact or inexact matching on any field. Duplicates can then be merged together according to some predefined merge rule or a custom merge rule. Merge rules specify how records are rolled up; which record is the master record in the system and which field data are selected in the case of conflicting data. Figure 3 shows the merge screen with two duplicate sets. Note that the first set contains two records, 8.1 and 8.2 with differences in account name, first name, and last name. Even with the different spellings the duplicates have been identified.



Dup?	Master Loc	Filter	CleanCRM.Account Name	CleanCRM.Contact First Name	CleanCRM.Contact La
Dup?	Master Loc	Filter	CleanCRM.Account Name	CleanCRM.Contact First Name	CleanCRM.Contact La
▣ 8.1	☐ CSV	▣ allow	Claus Productions	☐ Bob	☐ Beache
▣ 8.2	▣ CSV	▣ allow	Klaus Productions	▣ Robert	▣ Beach
Final			Klaus Productions	Robert	Beach
Dup?	Master Loc	Filter	CleanCRM.Account Name	CleanCRM.Contact First Name	CleanCRM.Contact La
▣ 9.1	☐ CSV	▣ allow	Sports-power	☐ Ývonne	Dwyer
▣ 9.2	▣ CSV	▣ allow	Sports Power	▣ Yvonne	Dwyer
▣ 9.3	☐ CSV	▣ allow	The Sports Power	☐ Yvonne	Dwyer
Final			Sports Power	Yvonne	Dwyer

Figure 3: Duplicate identification during batch processing.

A combination of domain knowledge and inexact matching allows for finding these records. The nickname ontology is deployed and in this instance finds that Bob and Robert refer to the same person. Note that if Bob or Robert had been misspelled, CleanCRM still would have found them as being duplicate because of inexact matching. The second duplicate set, with three duplicates of 9.1, 9.2, and 9.3 show how the robust inexact matching works including the handling of non-ascii character sets. The first names include Ývonne and Yvonne. In both duplicate sets a final record is shown that displays to the user what the final record will look like in the CRM system.

Data Quality Upon Migration

ActivePrime CleanMove is a data quality solution for assisting the migration of customer data from a previous, legacy CRM system to a new system. It expedites the data migration process by applying a series of data cleansing rules to customer data that verify the quality, integrity and format of the data. This includes verifying unique data in database key fields, referential integrity checking, phone, address, and date formatting, finding and merging duplicates, and other processing. Some CRM systems, like Oracle CRM On Demand, put strict controls upon the format of data before it can be imported. If data like phone, addresses and picklist values are not formatted properly, the database will reject the data. In other words, if the data is not clean then the customer can not even import the data. When millions of records are being brought into such systems, the value provided by CleanMove is very significant. Several months of either manual work or custom coding can be saved by using CleanMove.

The ActivePrime Data Quality Platform

All three ActivePrime products are built on the ActivePrime platform. This is a set of APIs, algorithms, 64-bit Linux servers, ontologies, specialized databases, and third-party data sets, all integrated to provide a platform for rapidly building data quality applications. For performance reasons low-level algorithms are coded in C or Cython, business logic, application logic, and ontologies are encoded in Python, and the user interfaces are coded in standards based, cross browser HTML and Javascript.

Customer Benefits

Over the past several years, experience at ActivePrime shows that the number of duplicate records in corporate CRM systems range from the low end of 10% to a staggering 70% on the high end. In addition it is common for at least 50% of CRM records to have faulty data, for instance, misspelled customer names or address data.

Using a combination of approaches, cleansing data upon initial migration and then ongoing cleansing real-time and in batch processes, duplicates can be drastically reduced, with typical deployments reaching below 1% duplicates and with a fix rate of up to 90% of faulty data.

Quantification of the business impact includes customers who have saved their sales representatives up to 4 hours per week. This savings are due to less time looking for the correct data, determining where the data should reside because of less duplicates, and then ease in communication with customers due to higher quality account and contact data.

For large companies, the savings of 4 hours per week per sales representative results in very large financial savings, and more important, efficient use of the sales force which directly results in more revenue. The combination of savings and extra revenue can quickly reach into the hundreds of thousands and sometimes millions of dollars.

Another significant customer benefit is enhanced reporting and analytics. For large public companies this is very important due the increase in securities compliances such as Sarbanes-Oxley. Failing to properly report on metrics such as customer counts and revenues has potential financial penalties that can be rather severe.

AI Techniques Used

ActivePrime leverages applied AI in three broad categories of search-space reduction (SSR) techniques, query optimization techniques, and ontologies. Before discussing the use of ontologies, the SSR and query optimization techniques are outlined.

SSR techniques are utilized when performing inexact matching on larger volumes of data, when record counts grow into the many thousands and millions. Robust inexact matching algorithms, comparing the closeness of two strings, have been in circulation for at least 40 years (Levenshtein 1965). The challenge today is how to perform such matching on larger volumes of data and very quickly. This is important in many data cleansing operations including normalization, searching for duplicates, and when performing record linkage (Dunn 1946). In such situations the combinatorial explosion of comparing each record in the large data set to every other record is prohibitive, especially given that each comparison operation itself is expensive. This is still an active area of research (Navarro 2001; Chattaraj, Parida 2005). ActivePrime technology leverages two types of SSR techniques; trimming of inexact comparison and inexact search indexing. Trimming is a one pass algorithm that scans over two strings, computing a histogram of the difference between the strings, and terminating if the total difference is above a distance threshold. If it does not terminate, then the full inexact comparison is performed. Tiered indexing is then performed on sets of strings, on multiple fields if possible. For instance, the indexing of first and last name when those two fields are to be searched. The sets can be indexed using several metrics, length being a common metric, and then searches over the indices are optimized. The order of scanning the index is adaptive, with the statistically fastest index scanned first, and then any logical AND based trimming of search terms is applied.

Query optimization techniques allow for high performance duplicate detection when comparing a small number of records to a large remote data base. This is deployed with ActivePrime CleanEnter. The user's query is analyzed based on the context of the fields, for instance company name or state name, and appropriate ontologies are referenced for the analysis. Through this analysis the query optimizer constructs a query that has a high probability of finding potential inexact matches while only retrieving a very small percentage of the remote database. Potential inexact matches are then quickly analyzed using SSR techniques.

Ontologies

A fundamental principle of AI application development is the separation of knowledge from algorithms that employ that knowledge. Through ontologies, users can express and share data management policies, such as data encoding conventions and context-specific interpretations. Decisions made during data acquisition, for example, can be saved for subsequent use, and propagated throughout the organization for coordinated data management. Employing ontologies to express general knowledge about data promotes flexibility in a data quality service, as solutions can be applied in a versatile manner across platforms, configurations, and data-management vendors. Interfaces that invoke the ontology deal with configuration differences, while fundamental domain knowledge remains unchanged. Past research in applying ontologies to data quality have focused more on abstract models for developing a framework for data quality (Wand, Wang 1996) and more recently to create ontologies for imperfect knowledge around mostly non-textual data, like GIS data (Frank 2007).

The past two decades have seen a broad range of projects leveraging ontology techniques. One such ambitious project is the Semantic Web, where the World Wide Web is extended with semantics, enabling machines to more easily reason about and leverage content and services. Three core technologies that are often associated with the Semantic Web are Extensible Markup Language (XML), Resource Description Framework (RDF), and Web Ontology Language (OWL). XML is the specification for creating custom markup languages and has found widespread adoption across virtually every domain in the computer industry. RDF, most commonly specified in XML, allows for conceptual modeling through the use of subject-predicate-object expressions. OWL, commonly serialized using XML RDF, is an ontology language for formally describing the meaning of terminology in web documents.

These Semantic Web technologies are very powerful and general purpose. That is both an enabler and an inhibitor; an enabler because of the breadth of solutions that can be considered using the technology and an inhibitor due to the complexity of the multiple layers and time and resources required to become very proficient in the technologies. Additionally, there are two other barriers that eventually led ActivePrime to look for alternatives. The first is the adoption rate of the technology. While there is reasonable adoption of RDF and OWL, it became clear to ActivePrime that other solutions would evolve over time, due to the various frictions in practically taking on the technology. The final barrier for ActivePrime is the ability for developers, and in some case even non-developers, to create, edit, and understand the representations stored in the documents. It is believed that the design of XML was overly focused on machine consumption, without a balanced approach that takes into account the human needs of the content. There are extensions or alternatives to the technologies, like N3, Turtle, and microformats, and ActivePrime is following these trends to understand if and when it may be the right time to adopt such technologies.

Object oriented programming (OOP) is one of the most successfully adopted paradigms in computing history, and OOP languages can provide a reasonable base upon which semantics can be built. This is even more so when considering that many OOP languages are a combination of paradigms including OOP, procedural and functional. Modern OOP languages provide more capabilities than are needed for representing ontologies, while at the same time they don't have predefined structures and mechanisms for broadly sharing semantic data. However, for limited circulation of semantic data, like deploying semantic data within ActivePrime products and services, OOP languages are more than sufficient. ActivePrime has chosen this path, to deploy ontologies with a general-purpose OOP language.

Lightweight Ontologies and Inexact Matching

For ActivePrime, the two main drivers for selecting an OOP language for representing ontologies were the ease by which programmers can create, edit, and manage ontologies, and the overall adoption of the language. One language that marks high on both counts is the Python programming language and is now the language of choice at ActivePrime for ontologies. Python is well known for its clarity and expressiveness and over the past decade has grown into one of the most popular new languages worldwide. In addition, several other capabilities like introspection, dynamic typing, and it being interpreted, has helped Python to be informally classified as a lightweight language. The lightweight nature of Python means that solutions can be quickly prototyped and then the prototype re-factored over time to evolve into a robust, production ready solution. For companies with aggressive goals, limited budgets, and evolving product specifications, lightweight languages are incredible enablers.

The ActivePrime Platform can work with any type of textual data and to date has focused on people-centric data. Financial, pharmaceutical, and custom ontologies are also deployed at customer sites, and these are used in conjunction with the standard people-centric ontologies. This is possible because of the clean separation of ontologies and computation. Ontologies can be easily plugged into the ActivePrime Platform, connecting the domain-independent data cleansing APIs and algorithms with the domain-specific ontologies. Over time, the repository of ontologies have naturally formed a hierarchy of specificity. The most general ontologies cover people-centric data elements. The next level comprises ontologies specific to particular vertical industries, such as financial services and pharmaceuticals. The most specific ontologies are customer generated. All ontologies are created by ActivePrime, with the custom ontologies being created as part of a short consulting engagement with the client. Future work at ActivePrime will include adding one more layer to the specificity, between industry specific and custom ontologies. This layer will allow third party system integrators to create their own ontologies on the ActivePrime Platform, giving them another competitive differentiator when going to market with their services. In addition to this new layer there will be intuitive user interfaces for allowing the non-technical user to create and manage the ontologies. System integrators, and in some cases the client themselves, will be empowered to create and manage their own ontologies.

Many people-centric ontologies are in use in the ActivePrime products. Seven such ontologies are outlined here.

- Nicknames identify various synonyms for people's first names. The ActivePrime products have nicknames for common American, Indian, and various European names. For instance Bob, Robert, Rob, Robbie, and Bobby are all synonyms.
- Company synonyms are also provided for the largest 1000 global companies. For instance, General Motors and GM are identified as synonyms.
- US postal code data is represented as relationships between the City, State, and Postal Code data, specifying which values correspond to valid US postal data. For instance, Pasadena, CA, 91102 represents a valid US postal record
- US and Canadian states, provinces, and territories are all represented in long, short, and abbreviated forms. For instance, California, CA, and Caly.
- Country names are all represented in long, short, and abbreviated form. For instance United States of America, USA, America, United States. Note that there can be multiple abbreviated forms.
- Business designations are also represented in long, short, and abbreviate form, also with multiple abbreviations if necessary. For instance, Limited Liability Corporation, LLC, Limited Liability Corp.
- City words are represented in long and short form. For instance Heights and Hgts.

The above ontologies combined with basic string search and replace is quite valuable for data cleansing operations.

However, the most significant value is realized when the ontologies are used with inexact matching. For instance, when normalizing country names to the long form, inexact matching can compare the incoming data with the country names ontology to obtain excellent results. The incoming data is compared to each country form, across the entire country ontology. Then the long form of the closest match is used. Table 1 below displays seven different spellings, some with errors, where the country matches to United States of America and is normalized to the short form, USA.

Incoming Data	**Matching Value**	**Match Type**	**Match Form**	**Result Data**
USA	USA	Exact	Short	USA
United States of America	United States of America	Exact	Long	USA
Amrika	America	Inexact	Abbr.	USA
America	America	Exact	Abbr.	USA
US of A	USA	Inexact	Short	USA
United States	United States	Exact	Abbr.	USA
Unitd states of Amerika	United States of America	Inexact	Long	USA

Table 1: Results of matching and standardizing to short form across the country ontology.

The countries can be stored in Python as either a list of lists or a dictionary of lists. Shown below is a dictionary of lists where each key is the long form of the country and the dictionary value is a list containing the short and abbreviated forms with each abbreviation stored as a separate entry in the list. Note that some countries may not have an abbreviation. The code below shows how the country list of lists is represented in Python, for brevity showing just five countries.

```
countries = {
  "Tuvalu":["TV"],
  "Uganda":["UG"],
  "Ukraine":["UA"],
  "United Arab Emirates":["AE","Emirates","UAE"],
  "United States of America":["USA","America","United
States"]
}
```

Benefits to Data Mining

Data mining uses have clearly expanded in reach the past decade, broadening out from the corporate business analytics system to now becoming a common component of data intensive, web based consumer and business systems (Segaran 2007).

When performing data mining there is often a requirement to perform analysis on numeric data with a dependency on associated textual data. For instance, when deriving analytics or performing pattern analysis on sales information in the 50 USA states, a significant issue that arises is simply normalizing the state names so that a small set of queries can be generated to obtain and prepare the data for analysis. If the data contains many different and unknown means of representing the state names, then the queries themselves become very complex. Worse than that, the result of the queries and hence the results of the analysis become suspect.

Robust data cleansing operations as described in this paper cab have a significant benefit in data mining applications. Note only can the analysis be more automated and hence operationally efficient, but the results themselves can be more readily accepted as valid results.

Engineering Lessons Learned

ActivePrime products connect to three CRM systems, by Oracle, Salesforce.com, and Sage SalesLogix, and are in use by over 150 customers, in over 40 countries, representing over 30 different languages. Companies as large as Fidelity and Oracle and as small as two person consulting firms use ActivePrime products. Several important lessons have been learned over the past few years regarding ontology representations, modifications, and storage.

Initial ActivePrime ontologies were stored in simple CSV files. A benefit from this simple format is that if necessary customers could easily modify the ontology to their specific needs without any custom or obscure ontology editor. As ActivePrime has moved towards more SaaS based solutions, the formats have been migrating to Python and if custom ontologies are required, ActivePrime sales engineers can assist with that. In the future, web based UI components will be provided to allow the customer to modify the ontologies. Modification will often be applied as an operation on the base Python code. The main ontology will stay intact and there will be code that modifies specific entries according to the customer's needs. For instance, the countries list can be customized with one line of code that adds another abbreviation to Ukraine.

```
countries["Ukraine"].append("Kievan Rus'")
```

An important observation is that Python can provide not only the static content for the ontologies but also operators for processing the ontologies, just like the append operation above. This opens up significant opportunities for dynamic ontologies. Ontologies are treated as OO classes where the class is mostly data, the ontology, and a small amount of processing. In many cases the class is simply an instances of a well known Python data structure, like the current ActivePrime ontologies, and then existing

Python operators, like the append operator above, are applied to specific pieces of the ontology. In other words, the Python class or data structure need not contain the actual functions for performing much of the processing.

Another advantage to Python is that if represented properly, data can be stored in JSON format, making it even more broadly appealing to the development community and making the transfer of ontologies amongst various technologies very easy. For instance, JSON is very easy to work with in Javascript as well as Python and other dynamic languages. There is less need for more complicated parsing technologies required by XML and XML based specifications. When speed and agility of development is important, formats like JSON provide clear and significant benefit. Some of the ActivePrime ontologies are Python code that is also valid JSON. In the future this may become a standard for all base ontologies.

Storage and versioning of ontologies has not been an issue yet. Ontologies are simply represented in plain text files as Python code and stored in a subversion repository. The repository provides versioning of the files. Beyond subversion versioning, a convention in the Python community is to put a __version__ global variable in each Python file. This stores information on the modules version. Note that modules in Python are the file itself, so myontology.py is the module named myontology. The convention is that __version__ is not updated every time the file changes, like the versioning information in subversion, but instead is updated to reflect the author's actual module's version when a new version is publicly released.

References

Experian QAS. 2006. U.S. Business Losing Revenue Through Poorly Managed Customer Data. *Experian QAS Briefing* 3/28/2006.

Dunn, H. 1946. Record Linkage. *American Journal of Public Health*, 36(12):1412-1416.

Levenshtein, V. 1965. Binary codes capable of correcting spurious insertions and deletions of ones. *Probl. Inf. Transmission,* 1:8–17.

Wand, Y,: Wang, R. 1996. Anchoring Data Quality Dimensions in Ontological Foundations. *Communications of the ACM,* 39(11):86-95.

Navarro, G. 2001. A Guided Tour to Approximate String Matching. *ACM Computing Surveys*, 33(1):31–88.

Chattaraj, A.: Parida, L. 2005. An inexact-suffix-tree-based algorithm for detecting extensible patterns. *Theor. Comput. Sci.* 335(1) :3-14

Frank, A.: 2007. Data Quality Ontology: An Ontology for Imperfect Knowledge. *8th COSIT International Conference*, 406-420.

Segaran, T.: 2007. Programming Collective Intelligence, 1st Edition. *O'reilly Media, Inc.*