ABSTRACT

Information on public safety is important to organized society for a number of reasons: First, citizens can use such records to identify risks in their communities. This can influence their short-term decisions (whether to visit a certain place or not) or long-term decisions, (such as whether to buy a house in a particular neighborhood). Second, government officials can use records on public safety and emergencies to make community-impacting decisions (such as budget adjustments for police forces). Third, information on safety-related activities can improve the transparency of, for example, law-enforcement organizations, enhancing their community relationship. In this paper we describe the creation of a website describing public safety events in the city of Troy, New York. Here, users can obtain information from different sources, to either browse on our website or use for their own services. Additionally, we have developed an ontology for representing these public safety events in a flexible and extensible manner. We describe the process and techniques used for integrating this information, the functionality made available through these approaches and the possible extensions to our work.

Categories and Subject Descriptors
H.4 [Information Systems Applications]: Miscellaneous; J.1 [Computer Applications]: Administrative Data Processing—Government

General Terms
Management, Experimentation, Human Factors

Keywords
Public Safety, Crime, Visualization, Data Integration, Semantic Technologies, Local Government

1. MOTIVATION

Since its inception, the Web has allowed people to interact and exchange information. On the Web, communities with different objectives and visions regularly emerge and evolve. Is in this scenario that the concept of “Social Machines” appears (i.e., mechanisms where human beings and the Web work together to solve challenging problems). There are several examples of incipient Social Machines (from Wikipedia to reCaptcha). A key objective in Social Machine research is to study such mechanisms and their properties. One way to do this is to create new Social Machines for solving a problem relevant to a community. One major issue facing the community of Troy, NY (the location of Rensselaer Polytechnic Institute) involves tracking and managing information on local emergencies. Thus, we sought to implement a Social Machine for achieving this aim, being the information integration and publishing the first step.

In general, the size and complexity of modern communities have increased the need for public safety information (for citizens, policy makers and safety enforcers). One problem with community-based exchange of such information is that non-verifiable information may be spread through social networks that may be uncertain, redundant or outdated. The accumulation of non-verifiable information by individual community members may result in misinterpretations of other information they encounter. As such, the lack of trustworthy information may lead individuals to wrongly evaluate situations and locations as “dangerous” or “non-dangerous”.

In order to provide information on emergencies having occurred in Troy, NY, we developed a website that allows individuals to find verifiable data on public safety events (e.g., assaults, fires, etc.) from trustworthy sources. This website seeks to answer questions such as “Is it safe to walk on 5th street between 4:00PM and 6:00PM, based on recent events?”, “Can I park my car in this neighborhood without a high risk of it being stolen?”, or “How safe is the area around the school that my children attend?”. The answers to these questions depend on what sources of information an individual trusts: In all likelihood, an individual will be skeptical of information from certain sources, while finding other information sources trustworthy. Hence, metadata on the source(s) of information can serve as an important aid in establishing trust.

The problem of integrating and publishing information related to public safety is interesting and worthwhile to pursue for several reasons. First, access to such information directly benefits individuals of a community, as they can use such records to identify risks in their communities. Second, allowing individuals to identify which sources of information to trust can enhance their confidence in acquired knowledge. Inherently, information from a set of sources deemed trustworthy can work together to solve challenging problems.
trustworthy by an individual may not provide an ultimate truth. However, it can certainly allow them to form their own opinions, as they may from reading trustworthy newspapers. Third, the challenge of displaying information on past emergencies (in a way that is attractive and informative at the same time) is critical; for an individual to effectively interact with information, usability-related issues must be kept in mind.

2. BACKGROUND

United States Government data published on the Web has helped to increase the transparency and accountability of different organizations and agencies[1][11][25]. The enhanced visibility of this data has motivated citizens to create applications (usually Web-based) that use it in multiple ways. Thus, it is possible to compare voting records of US senators[28] with supplemental information, such as political affiliations. Other applications are aimed at showing nationwide data on different issues, such as the environment and economy. One such application shows the ozone levels across different regions of the US, using datasets published by the Environmental Protection Agency (EPA)[9]. Similar government initiatives to publish data have been taken by other countries such as Canada, the United Kingdom, Australia and Greece. There have been several initiatives to promote the use of so-called “Open Government Data”, where citizens are encouraged to come up with new applications for government data: Events like “The Great American Hackathon” sponsored by the SunlightLabs[33] in the US and “GovHack”, sponsored by Government 2.0 Task Force in Australia[17] recruit developers for creating applications based on one or more available government datasets.

2.1 Related Work

The idea of crime data analysis is not new, with several examples of relevant prior work[19][18]. At the time we developed our website, we found a similar project called Every-Block[12], which provides local information related to specific cities, like New York, Boston and San Francisco. This information is taken from local government offices and covers different topics, depending upon information availability from each city.

Another example is present in several major cities, where police departments have developed their own geographical information systems for their citizens: This is the case of ClearMap[5] in Chicago, Illinois and CrimeMapper from Portland, Oregon. The main goal of these projects is to map crimes related to a specific areas and neighborhoods. While most of these projects are focused on providing valuable information to citizens, there is no easy way to manage the data in different ways from the ones intended by their creators. For example, it is not easy to merge or compare information about specific crimes between cities or create new applications that use information provided by these websites in a easy way. This is mainly because most of these websites do not allow users to manage the “raw” data but only visualize it through their interfaces.

From the perspective of knowledge representation, several approaches have been developed. Most of these developments have focused on fully modeling a legal framework or defining rich ontologies[24][2][22][15]. Our needs pointed to a simpler but more specific solution, where geolocation information was a priority. Also, given the necessity of integrating governmental information from police departments and universities, we needed to create ontology aligned with the legal definitions provided by the Federal Bureau of Investigation (FBI)[13], capable of being extended later. Through extensibility, it becomes possible to use this ontology with other institutions and police departments as well.

2.2 Publishing Mechanisms for Government Data

Government agencies have published data on the Web for years. In May 2009, President Obama’s administration created Data.gov, a central repository of information generated or collected by the Federal Government. As of this writing, it is possible to find datasets covering a broad range of domains, from agriculture to financial and military areas. This data is published on the Web mainly using eXtended Markup Language (XML), Microsoft Excel (XLS), Comma-Separated Values (CSV), Keyhole Markup Language (KML or KMZ) and Shapefile (SHP) file formats. One of the benefits of these formats is the huge number of libraries and applications that support them. This gives developers multiple options to process, interact with and visualize the data. On the other hand, it becomes a non-trivial task to mash several datasets to be processed or visualized. This is important for finding possible correlations (or potential causality) among variables that are described by different agencies (e.g., find possible correlation between children’s health problems and level of pollution on a specific location). The lack of a common language that can be used to express information makes it difficult to use different datasets. Currently, there is no clear and easy way for machines to represent entities such as people, cars and years.

In recent months, the Tetherless World Constellation at Rensselaer Polytechnic Institute has been[10] converting the datasets available at Data.gov into RDF (Resource Description Framework). The goal of this effort is to provide multiple datasets in a common format that makes them easier for developers to reuse the data in a Web-based environment. A similar approach has been taken by the Government of the United Kingdom, where data “will be published using open standards and following the recommendations of the World Wide Web Consortium” and datasets “will be represented in linked data form”[21].

2.3 Leveraging Data Usability with Semantic Technologies

One of the goals of the Semantic Web is to make machine-understandable data available on the Web. This means that artificial agents (i.e., computer programs) will be able to use the data and take corresponding actions. While HTML (HyperText Markup Language)[6] can describe elements in web pages, there is a need for describing other entities, such as people, cars or events. This led to the creation of Resource Description Framework (RDF)[29], a standard model to describe resources on the Web. According to RFC3986[3] specification, the word resource “is used in a general sense for whatever might be identified by a URI. Familiar examples include an electronic document, an image, a source of information with a consistent purpose (e.g., “today’s weather report for Los Angeles”), a service (e.g., an HTTP-to-SMS gateway), and a collection of other resources.”

Using RDF, it is possible to express specific information about resources (e.g., Alice’s phone number) and relation-
ships between them (e.g., Bob is Alice’s brother). More expressive languages have been developed based on RDF, such as RDF Schema (RDFS)[30] and the Web Ontology Language (OWL)[27], which allow information to be defined in greater detail, using concepts like Classes and Subclasses (in RDFS) or Description Logics (in OWL DL). In order to define concepts in specific knowledge domains, people started creating ontologies (i.e., formal representations of concepts and relations between them). These ontologies are usually expressed in RDF, RDF Schema and OWL and serve as a basis for people to express information that can be published, shared and mixed in an easy way. Thus, it is possible to find ontologies for describing social networks (FOAF)[14], representing genetic information (GeneOntology) or characterizing online communities (SIOC). These ontologies help communities to express information about specific areas of knowledge, lowering the costs of collecting, exchanging and using information.

## 3. DATA INTEGRATION

In this section we present the work done for translating, curating and publishing public safety data in RDF. Also, we explain the methodologies and techniques applied to develop our emergency tracking website. We based our work on two different sources of information: reports from the Rensselaer Polytechnic Institute Public Safety department (from now on RPS data) and information from the Troy Police Department (from now on TPD data).

The conversion process consisted of the following steps: (i) searching for, and downloading, the files, (ii) extracting desired information from them and (iii) creating instances for each event and adding the properties and curated relations to each of them using the ontology mentioned above.

### 3.1 Ontology development

We decided that in order to integrate Troy Police Department and RPI Public Safety data, we needed a lightweight ontology to include the events described in both datasets. As mentioned in Section 2.1, existing work include ontologies that represent a more complex scenario than the one we faced. Moreover, geographical information is not a priority in these ontologies. Thus, we created a small taxonomy of public safety events. Some of the classes can be seen in Figure 1. We divided Events into four major categories: Non-offense, Offense, False and Others. We based the Offense Events on the “Offenses Known to Law Enforcement”[13] description provided by the Federal Bureau of Investigation (FBI): This include description of violent crimes (forcible rape, aggravated assault) as well as property crimes (robbery, burglary). Since RPS data contains other events that cannot be classified as “crimes” (e.g., An accident, or a fire) we created a Non-offense events category, which includes events such as “Medical Events”, “Fire Events”, and so on. Finally we realized the existence of two types of events in RPS data that cannot be classified as a subclass of the events described previously: False events and Other events. Hence, it was necessary to add them to the taxonomy. In particular, “False Event” contains several subcategories, such as “False

![Figure 1: Part of the taxonomy that describe public safety events. Not all events are shown here.](http://www.rpi.edu/dept/public_safety/)

Alarm” and “False Fire”.

We also defined a class to represent a “Source Of Information”. We created two instances of this class: One for RPS data and other for TPD data. In this way, it is always possible to add more sources without the need to change part of the ontology itself: It is only necessary to add new instances as a source of information and entering new events related to that source. Finally, every event is related to several (typed) values, such as its geolocation, date of reporting, disposition and approximate location (such as a specific building or address). The ontology is available at http://publicsafetymap.org/ontology.owl.

### 3.2 Data Conversion

For the case of TPD data, we received an Excel file containing addresses (or street intersections) in natural language, date and time of the event and the type of the event. Converting it to CSV file and parsing the data using Perl scripts was a straightforward task.

The case of RPS data was more complicated. First, RPS data is published in the website of the Rensselaer Polytechnic Institute Public Safety website in PDF files. Prior to 2008, the format of publishing contained several fields: A code number, date, time, a location, which described the building or area closest to where the event happened, a code for the type of event, a boolean value to indicate whether it was a “hate crime” (i.e., a crime motivated by race or other social factors) and a description of the disposition or action taken. From 2008 onward, and for every monthly report from the year 2009 onward, the layout as well as the information provided changed: Each event was described by an event number, a date and time of the report, date and time when the event started and date and time when it finished, a type of event (not consistent with previous format) and a report number (consistent with the previous code number).

During the conversion process, every event was identified as an instance of the classes "False Event" and "Offense Event". The conversion process consisted of the following steps: (i) searching for, and downloading, the files, (ii) extracting desired information from them and (iii) creating instances for each event and adding the properties and curated relations to each of them using the ontology mentioned above.

---

2As an example, it is possible to think of a program that collects information described using a specific set of ontologies, stores it and analyzes them, without a human to interfere.

3The curation is described in section 3.3.
by a resource. We decided to reuse a predicate from RPI Map\(^5\) “Has\_LatLong” to include geolocation information in WGS84 standard\(^7\). For the dates we considered the typed literals as xsd:date\(\text{Time}\) (a XML datatype). For natural language information we considered them as xsd:string. Even though the report Id of each event can be considered as a numeric value (or more precisely a date), we decided to consider them as string, due the fact the it is possible that a sample of the data converted can be seen in Figure 2.

### 3.3 Data Curation

Using TPD\(\text{data}\) and RPS\(\text{data}\) was not enough to specify the position (i.e., geolocation) where the events occurred. In order to achieve this, we obtained geographical coordinates from the Google Maps API\(^6\). This service takes a string describing an address or location (i.e., “4th street and Broadway”, “742 Evergreen Terrace”) and returns a JSON\(^6\) object which contains a coordinate tuple (based on WGS84). Since the input may be ambiguous or too extended (e.g., “2nd street”), Google Maps returns also an “accuracy” value, ranging from 0 to 9: the higher, the more precise the geolocation is.

On the other hand, since the classification of the events was not clear, it was necessary to consolidate them under the ontology described in Section 3.1. This step required several iterations of human-driven consolidation due the different versions for classifying specific events (the code was different in pre-2008 format and post-2008 format) as well as typos in the strings that described the type of event.

### 4. IMPLEMENTATION

The development of the website was done using PHP 4.4.9 as a server-side script language and JQuery\(^31\) 1.3.2 and JQueryUI 1.7.2 for client-side scripting. For visualizing data, we used the Google Visualization API and MIT Exhibit\(^23\). Both these tools take input data in several formats (JSON, XML, others) and visualize it through a faceted browsing interface, region map, and an annotated timeline. We show some of the visualizations used in Figures 3 and 4.

#### 4.1 Data Persistence

Data storage was one of the main technical issues we faced. Since the amount of data involved was small (less than 200,000 triples which is 50MB approximately) we did not want a highly scalable triple store (with all the costs in time and resources involved). Instead, we decided to use ARC2\(^26\), a PHP library for managing RDF. ARC2 includes a SPARQL endpoint that can use a MySQL or PostgreSQL backend and also support aggregation functions (count, group) and SPARUL\(^32\) instructions for inserting and deleting triples. One of the advantages of ARC2 over other triple stores is its ease of use: A SPARQL endpoint can be configured on any LAMP\(^7\) server in few steps.

#### 4.2 Visualization and Querying

Searching events can be done in several ways. First, it is possible to select which sources of information the user wants to consult. This is important, because it gives users the ability to choose which sources they trust. Thus, a person not affiliated with Rensselaer Polytechnic Institute may consider that only a city Police Department is a valid and trusted information source and hence only consider events from TPD\(\text{data}\). Second, users can query for specific events or aggregated results (e.g., “Return only events related to Property Crimes”). This allows users to obtain a better “big picture” of specific types of events (such as Property Crimes) without displaying other events that may cause confusion.

Third, it is possible to select the time frame for events to be displayed. On our website, it is possible to select events ranging from only occurring 24 hours ago, up to 4 years ago. This can help users visualize short term as well as long-term trends. For example a user can filter certain events (e.g., robberies) that occurred at a specific time (between 5 and 7PM) in the last 3 months so she can have an idea of where the hotspots are that she should avoid.

Since we are showing multiple events from different categories, we decided to present them in two ways. First, as seen in Figure 3, we used the Exhibit\(^23\) faceted browser. Here, we can map events using geographic coordinates added in the curation step. Users can filter results by several criteria: type of event, date of occurrence, time occurrence, source of information, and building or address location (when available). Second, we present to the user an aggregated timeline using the Google Visualization API as seen in Figure 4. Here, the user can see how many events of the selected type occurred everyday. It is also possible to zoom in to a specific day, week or month.

#### 4.3 Publishing and Exporting data

Although visualizations are useful for humans, we do not have a trusted information source and hence only consider events from TPD\(\text{data}\). Second, users can query for specific events or aggregated results (e.g., “Return only events related to Property Crimes”). This allows users to obtain a better “big picture” of specific types of events (such as Property Crimes) without displaying other events that may cause confusion.

Third, it is possible to select the time frame for events to be displayed. On our website, it is possible to select events ranging from only occurring 24 hours ago, up to 4 years ago. This can help users visualize short term as well as long-term trends. For example a user can filter certain events (e.g., robberies) that occurred at a specific time (between 5 and 7PM) in the last 3 months so she can have an idea of where the hotspots are that she should avoid.

Since we are showing multiple events from different categories, we decided to present them in two ways. First, as seen in Figure 3, we used the Exhibit\(^23\) faceted browser. Here, we can map events using geographic coordinates added in the curation step. Users can filter results by several criteria: type of event, date of occurrence, time occurrence, source of information, and building or address location (when available). Second, we present to the user an aggregated timeline using the Google Visualization API as seen in Figure 4. Here, the user can see how many events of the selected type occurred everyday. It is also possible to zoom in to a specific day, week or month.

![Figure 2: Representation of a medical-related event in RDF/XML format.](image-url)
want to limit the possibilities of the data. In this sense we allow users to export the “raw” data from our website in two different ways: RSS and RDF. We created an RSS feed for every search a user queries, thus making it possible for users to track new events in a regular RSS feeder (e.g., “I want information exclusively about robberies occurred in the last 7 days”) instead of being forced to use our interface. Our RDF exporting approach is similar to RSS: it is possible to retrieve triples based on a query specified by the user. It is also possible to retrieve all the events until a certain limit defined in order to avoid overload in the backend. Using any of both approaches, it is possible to reuse the data in other applications or merging it with additional data form other sources.

5. CONCLUSIONS AND FUTURE WORK
In this paper we described the development of a website for querying, discovering, visualizing and exporting data related to local public safety events. This project is part of a recent trend involving the application of semantics to government data. We listed a set of specific steps, showing (i) the conversion of data, (ii) the curation and (iii) inte-
guration using semantic technologies. We also justified the need for a simple ontology that allowed us to describe public safety events and information related to them, including where they occurred. Furthermore, we discussed different aspects of implementation (architecture and tools employed) as well as the decisions involved in designing the website and modeling the data.

As mentioned in Section 1, we see two different problems to solve. Our short-term goal is to publicize our website through different channels to make it available for the greater web community. This lead to several practical problems, such as automating data acquisition and adding more official data sources (for example, events reported to police departments in nearby cities or universities). While these tasks are not particularly challenging from a research point of view, they can have a strong impact on the value perceived by the users.

Our long-term objective is to study the properties of “Social Machines” (describe in Section 1). An interesting feature we have developed (but not released yet) is to allow users to report events directly to our website. This initiative raises several issues: How these events can be confirmed? How can we deal with privacy issues when a user is reporting an event? How is it possible to detect events that have been reported twice? How can people make sure this information is trustworthy? Several mechanisms should be developed to ensure these questions can be answered satisfactorily.

As mentioned above, there is an interesting opportunity to add value for users integrating datasets from other organizations, such as the Fire Department, or Police Departments from nearby cities. The use of semantic technologies would simplify the addition of these datasets into our repository, in the same way that helped integrating RPS data and TPD data. Another idea for further development is to give users extended options for search; while we have shown this demo to several people, more user feedback is needed to develop these extended functionalities. Finally, the possibility of exporting data in RDF and RSS allows third-party developers to create their own applications. This increases the value of the available data making it useful for other purposes that these developers may find interesting.

5.1 Acknowledgments

We would like to thank Anson Byun, Pavan Rao, You Li, Yixi Zhang and James Michaelis for helping in this project and paper.

6. REFERENCES


[26] B. Nowack. Arc, rdf classes for php

[27] Owl web ontology language overview,

[28] Participatory Politics Foundation and Sunlight Foundation. Open congress, head-to-head voting comparison,


[30] Rdf vocabulary description language 1.0: Rdf schema,

[31] J. Resig. jquery: The write less, do more, javascript library,

http://www.w3.org/Submission/SPARQL-Update/, 2008.