Ontology-Centered Syndromic Surveillance for Bioterrorism

Monica Crubézy[1], Martin O'Connor[1], David L. Buckeridge[1,2], Zachary Pincus[1], Mark A. Musen[1]

[1]Stanford University, Stanford, CA 94305-5479
[2]VA Palo Alto Health Care System, Palo Alto, CA

Abstract

Syndromic surveillance requires the acquisition and analysis of data that may be suggestive of early epidemics in a community, long before there is categorical evidence of unusual infection. These data are often heterogenous and noisy, and need to be interpreted by a combination of analytic methods. Syndromic surveillance thus involves problems of integrating data, of configuring problem-solving strategies, and of mapping integrated data to appropriate methods. These tasks have been studied in the knowledge-based systems community for many years. We present a software architecture that suppports knoweldge-based data integration and problem solving, thereby facilitating many aspects of syndromic surveillance. Central to our approach, a set of reference ontologies supports semantic integration and a parallelizable blackboard architecture implements invocation of appropriate problem-solving methods and control of reasoning. We demonstrate our approach with BioSTORM — an experimental system that offers an end-to-end solution to syndromic surveillance.

Keywords: Ontologies; knowledge modeling; knowledge-based systems; ontology mapping; data integration; problem-solving methods; syndromic surveillance; bioterrorism tracking, alerting, and analysis; disease prevention and detection.

A New Trend: Syndromic Surveillance

In recent years, public health surveillance has become a priority, driven by concerns of possible bioterrorist attacks and disease outbreaks. Authorities argue that early detection of nascent outbreaks through surveillance of "pre-diagnostic" data is crucial to prevent massive illness and death (Pavlin 1999). This need, and the increasing availability of electronic data, have resulted in a blossoming of surveillance system development (Bravata et al. 2004). Most recent systems use electronically available data and statistical analytic methods. In general, the emphasis is on interpreting noisy, non-definitive data sources, such as diagnosis codes from emergency room visits, reports of drug sales and absenteeism, calls to medical advice personnel, etc.

For example, the Real-time Outbreak Detection System (RODS; Tsui et al. 2003) allows automated transmission and analysis of diagnostic codes and other data from hospital information systems at many emergency rooms in the Pittsburgh area. More recently, the U.S. Centers for Disease Control and Prevention (CDC) began developing the BioSense system, which monitors data from many sources, including U.S Department of Defense and Veterans Affairs (VA) facilities, and over-the-counter pharmaceutical sales. By the summer of 2003, public health authorities already had deployed more than 100 different surveillance systems in the United States, all relying on electronically available data to detect disease outbreaks rapidly (Buehler et al. 2003).

The increasing availability of electronic data within the public health information infrastructure presents tremendous opportunities for surveillance and, at the same time, considerable problems. The opportunities include improving public health decision making by extracting more information from a growing set of integrated data sources. The problems include technical barriers to incorporating physically heterogeneous data sources into a surveillance system and, more importantly, the difficulty of integrating disparate data sources in a way that can offer semantic coherence and improve decision making.

In most situations, electronically available surveillance data are not collected for the expressed purpose of monitoring public health. Recently deployed surveillance systems tend to rely on data collected for administrative and business purposes. For example, many systems follow records collected to enable billing or pharmaceutical

sales records collected for inventory and marketing purposes. Because these data sources are not collected with surveillance in mind, they are often biased. In addition, because public health agencies do not control data collection, data rarely conform to a standard format. When incorporating data sources into a surveillance system, differences in representing biosurveillance concepts must be reconciled. *Semantic reconciliation* is especially important so that analyses across data sources can integrate conceptually diverse data and then reason about the data in a consistent manner. Unfortunately, existing surveillance systems do not have comprehensive data-integration strategies; instead, they are developed specifically to operate using the small number of data sources available at the time of system implementation. These systems are therefore extremely limited, in terms of the usable data and by the significant effort required to incorporate novel data streams.

Surveillance systems also have complex operational and research requirements. The straightforward time-series algorithms used for summarizing traditional surveillance data are not suitable for integrating the complex data processed by modern surveillance systems. Surveillance data are rich in spatial and temporal measurements, which need to be aggregated and analyzed in meaningful ways. Thus, the high dimensionality, heterogeneity, and unpredictable nature of both data and disease-outbreak patterns all require that systems have a range of analytic methods which can make sense out of surveillance data in a wide range of situations. A variety of methods are required to process large volumes of data, to identify weak signals, to analyze multiple indicators, and to account for spatial structure. Moreover, these different methods must be used together in potentially complex configurations to provide appropriate results.

The process of analyzing surveillance data for its interpretation thus becomes a *problem-solving* process that involves a set of methods suitably chosen and tailored to each situation at hand. Rather than implementing a specific and *ad hoc* approach, surveillance systems need to provide an infrastructure for applying different analytic strategies on incoming data streams. The current approach to incorporating algorithms into surveillance systems, however, is to hand-tune a small set of analytic methods to one or two single data sources. This type of solution is inadequate for assembling and evaluating different analytic strategies without substantial reprogramming. In particular, this approach does not address either the possible addition of new types of data into the system or the crucial need for experimentation and configuration of analytic methods to help public-health officials in their outbreak-surveillance tasks. Next-generation surveillance systems should accomodate the high dimensionality and the semantic richness of surveillance data, provide a means to reconcile its semantic heterogeneity, and support the wide range of analytic methods necessary for data interpretion.

Ontology-Centered Syndromic Surveillance

To meet the complex operational and research needs of syndromic surveillance, and as part of DARPA's national program in biosurveillance technology, we have developed BioSTORM (Biological Spatio-Temporal Outbreak Reasoning Module). BioSTORM is an end-to-end computational framework that combines a variety of data sources and analytic problem solvers with the goal of meeting the performance demands of emerging disease-surveillance systems.

A central aspect of our approach is the use of ontologies to model and annotate information and knowledge involved in syndromic surveillance. Ontologies are computer-stored specifications of concepts, properties and relationships important for describing a domain of expertise. They provide principled, structured, and queryable frameworks for modeling the semantics of knowledge and data and encoding them, independently of the way they are internally represented. Ontologies enable the description of characteristics, types, and relationships of data emanating from different sources (Pincus and Musen, 2003), as well as the specification of repositories of methods for problem solving (Crubézy and Musen, 2003). Ontologies are used widely in the computer-science community to aid semantic integration of disparate data sources. In recent years, the biomedical community has adopted the approach at large, and some ontologies are central to information systems, such as UMLS and SNOMED.

The adoption of an *ontology-centered* framework for integrating surveillance data and processing them provides a durable foundation for principled, reproducible, and scaleable implementation and evaluation of approaches to public health surveillance. We have been working for nearly two decades to build intelligent systems that rely on ontologies to model and operationalize their different components. One of the major results of this research is

Protégé (Sidebar 1), a mature methodology and software tool for building ontology-centered, component-based systems. Protégé was the basis for all ontologies underlying the components of the BioSTORM system.

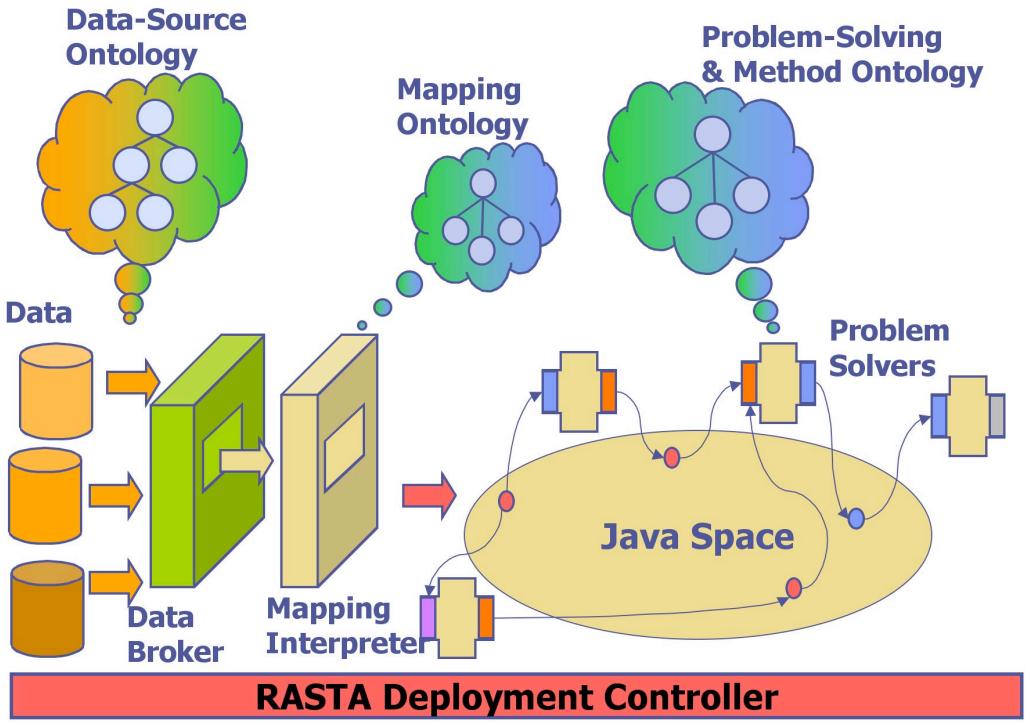

Figure 1. Overview of BioSTORM architecture and components. The RASTA controller orchestrates the deployment of problem solvers and the flow of data to them. Incoming data streams are passed through the Data Broker and Mapping Interpreter to a set of problem solvers. The Data Source and Mapping ontologies are used by the data broker and the mapping interpreter to construct semantically uniform streams of data for the deployed problem solvers. RASTA uses the Problem-Solving and Method ontology to configure sets of problem solvers into analytic strategies that operate on those data streams. The deployment of problem-solving strategies involves a Java Space blackboard mechanism.

Figure 1 shows the BioSTORM system's four main components: (1) a **data-source ontology** for describing the features of data sources and data streams; (2) a **library of statistical and knowledge-based problem solvers** for analyzing syndromic surveillance data; (3) an intelligent mediation component that includes (a) a **data broker** to integrate multiple, related data sources described in the data-source ontology and (b) a **mapping interpreter** to connect the integrated data from the data broker to the problem solvers that can best analyze those data; and (4) a **control structure** (RASTA), that deploys configurations of problem solvers to analyze incoming streams of data.

The novelties of our solution are (1) the central use of ontologies, which underlie each component of our surveillance architecture and allow the system to encompass the rich semantics of incoming surveillance data as well as of analytic problem solvers; (2) the companion use of a two-tier mediation component, which allows the system to reconcile the heterogeneous semantics of surveillance data with those of a variety of reusable analytic problem solvers. As a result, our approach leads to the meaningful configuration and the flexible deployment of knowledge-level surveillance-analysis strategies.

We successfully used BioSTORM for surveillance analyses of dispatch data from the San Francisco 911 emergency call center and Palo Alto Veteran's Administration (VA) Medical Center. We used this data to determine if patterns in 911 dispatch and ER data could be used to detect large scale influenza-like outbreaks (Sidebar 2). In addition, emergency room respiratory records in the Norfolk, Virginia area were analyzed in space and time, primarily to demonstrate the ability of our system to deploy multiple distinct analytic strategies in parallel on large data sets.

A Data-Source Ontology for Describing and Contextualizing Data Streams

Public health surveillance data are diverse and usually distributed in various databases and files with little common semantic or syntactic structure. To enable analyses using disparate data sources, knowledge about how to characterize and combine different sources and types of data must be specified precisely. In order to apply appropriate analytical methods to the relevant data for outbreak detection, data sources must be related to one another, to descriptions of reportable conditions, and to enumerations of the primitive data on which diagnoses of those conditions can be made. This general knowledge is best modeled with ontologies. We have developed a **data-source ontology** (Pincus and Musen, 2003) that provides a means for describing extremely diverse data in a coherent manner. The ontology also facilitates reasoning with and processing of the data. Further, our ontology can be customized for a particular domain of data sources, such as syndromic surveillance.

Our data-source ontology aims to make data self-descriptive by associating a structured, multi-level *context* with each potential data source. A developer describes the context from a data source by filling in a template with details about it, at the levels of data source, groups of related data, and atomic data elements (Figure 2, left). Our data-source ontology, as customized for syndromic surveillance, provides a taxonomy of data source attributes to describe this context (Figure 2, right). Developers describe individual data elements with metadata terms adopted from the Logical Identifier Names and Codes (LOINC). The LOINC approach describes a piece of data along five major semantic axes, including "kind-of-property," "time aspect" and "scale" axes. Clinical pathologists use LOINC to contextualize results reported by clinical laboratories. We have generalized the LOINC axes from this role into a generic set of descriptors for contextualizing many different types of data involved in syndromic surveillance. Our five axes are: (1) What is being measured? (*e.g.*, "Robitussin sales"); (2) How is it measured? (*e.g.*, "Cases sold per day") (3) When/For how long? (*e.g.* "Averaged over a week"); (4) Where? and (5) What are the possible values?

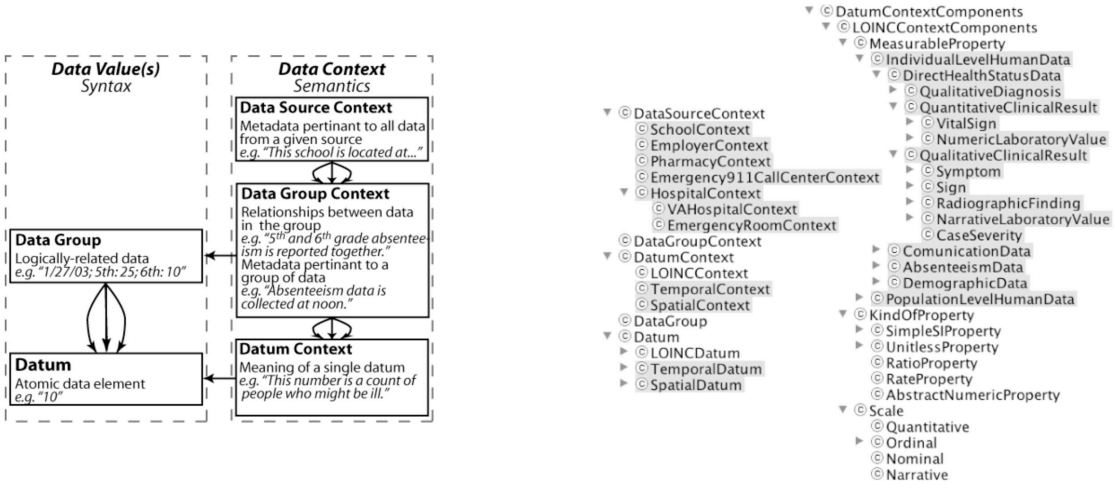

Figure 2. Template data-source ontology for data and metadata, customized for syndromic surveillance. Left, data values are associated with metadata describing the data and other relevant context. Arrows indicate one-to-one and many-to-one relationships between concepts. Right, highlights show additions to our template data-source ontology (developed in Protégé) that are specific to syndromic surveillance. The first snapshot shows the structure of the template with our added context classes. The second snapshot shows the top levels of the taxonomy of metadata attributes, expanded with our vocabulary of "Measurable Properties," used to build LOINC objects.

In our work with San Francisco Emergency 911 dispatch data and patient data from the Palo Alto VA medical center, as well as data related to reportable diseases, the data-source ontology was able to capture individual-level primitive data (signs and symptoms, laboratory tests) as well as observable population-level data (aggregated syndrome counts, school absenteeism). The ontology offers descriptions of generic data sources such as 911 dispatch data, with instances of the generic descriptions that describe the specific data fields of particular data

sources (e.g., the 911 dispatch data available in San Francisco). Describing the 911 and the VA data sources merely required completing the templates provided by our data-source ontology by selecting appropriate properties for describing the type of data emanating from each of these data sources.

The systematic, template-directed process supported by our data-source ontology allows developers to create a customized local model of each surveillance data source. Each local model shares a common structure, space of attributes, and set of possible attribute values. Without discarding the specificities of each data source, our data-source ontology provides a semantically uniform representation of each one. More precisely, the ontology provides a hybrid approach to data integration, in that it combines the semantic rigor of a global, shared ontology with the flexibility and level of detail that comes from devising customized, local ontologies for each data source. Most importantly, the ontology provides an abstract, metadata-rich view on data sources, that is unconcerned with the way data are stored (tab-delimited files, XML documents, etc.). This metadata thus supports the integration of heterogeneous surveillance data at the level of semantic reconciliation, allowing uniform application of analytic problem solvers to each data source.

A Library of Surveillance Problem-Solving Methods

Next-generation surveillance systems require a variety of analytic methods, ranging from traditional statistical techniques operating on low-level data such as raw disease counts, to knowledge-based approaches capable of reasoning about qualitative data and unusual patterns. In addition, systems must be capable of making correlations among different kinds of data, and must be able to aggregate and abstract data into information about populations, spatial regions, or temporal intervals. As a result, analyzing surveillance data is best seen as a problem-solving task addressed with a set of problem-solving methods carefully configured to the case at hand. BioSTORM has a library of computational methods that address the analysis of multiple, varying types of data for detecting this problem (Buckeridge et al. 2003). BioSTORM's library includes both generic, disease-independent statistical methods that analyze data as single or multiple time series, and knowledge-based methods that relate detected abnormalities to knowledge about reportable diseases. Relatively straightforward methods are implemented as simple software routines, while complicated methods are incorporated into the system by "wrapping" existing software libraries so that they conform to our method ontology requirements (see below).

In addition to syndromic surveillance, a system must also model performance characteristics, data requirements, and assumptions of each method. The methods in our library are categorized by the tasks that they perform, the data types that they can operate on, and the signal types that they can detect. For example, a *cumulative sum* method signals an abnormality in a single temporal data stream and is well-suited to detecting gradually increasing signals. Making such knowledge explicit facilitates system modification, both to enhance portability and reuse of methods, and helps public health professionals to understand the system. Naturally, ontologies are ideally suited to providing the modeling framework necessary to categorize and annotate collections of problem solvers.

Our approach was motivated by task-analysis approaches developed in the knowledge-based systems community, and was guided by the UPML framework for modeling libraries of problem-solving methods (http://www.cs.vu.nl/~upml/). Such a framework involves modeling the system's top-level task (monitoring of plublic health data), and identifying a problem-solving method that can perform that task (a method called surveillance). The method can be modeled as entailing a number of subtasks, and each subtask can, in turn, be solved by a problem-solving method which may entail new subtasks (Figure 3). Modeling continues until there is a primitive method that can solve each of the subtasks (*e.g.*, an autoregressive integrated moving average (ARIMA) temporal statistical algorithm). This model has been very helpful even when many of the surveillance methods are statistical, rather than knowledge-based.

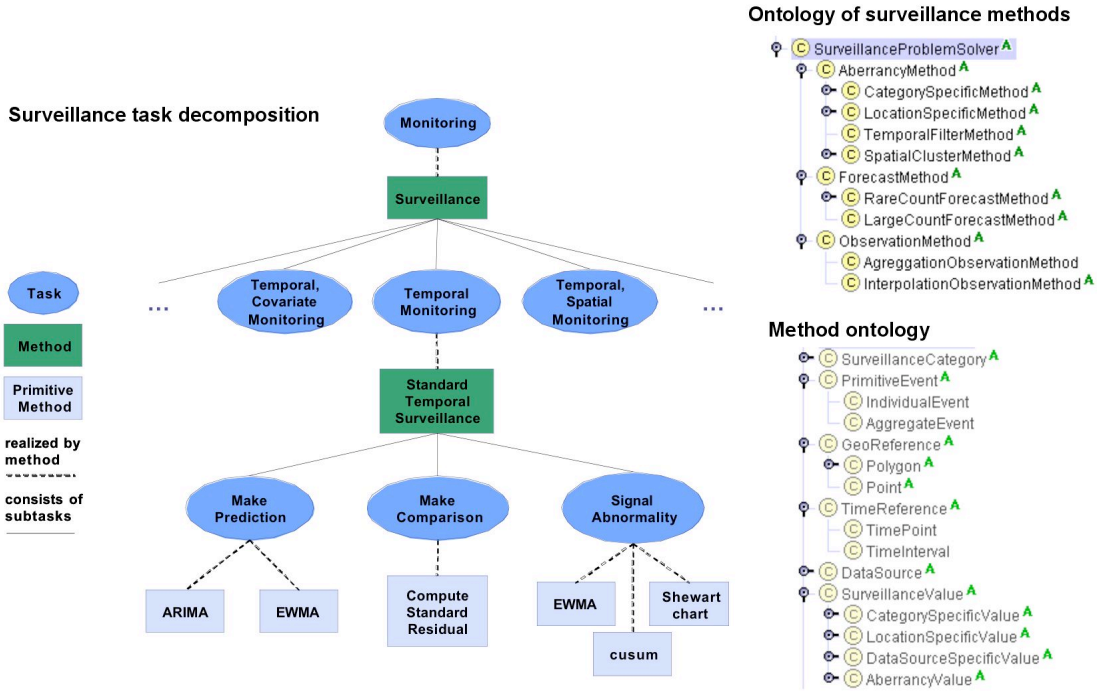

Figure 3. The surveillance problem-solving ontology. Left: a schematic representation of the refinement of the monitoring task into sub-tasks defined by information scenarios in public health surveillance, and the decomposition of standard temporal surveillance into further sub-tasks. Top right, a Protégé snapshot of our resulting surveillance problem-solving ontology, focused on the *ontology of surveillance methods*. Bottom right, another Protégé snapshot of the problem-solving ontology, focused on the *method ontology* in which each problem-solving method declares its expectations on input and output data.

We developed a practical, ontology-based framework for categorizing abnormality-detection algorithms. This framework is based on the information contexts commonly encountered in surveillance work and on the functional requirements of the algorithms (Buckeridge et al., 2003). We created it by modeling the overall surveillance task as a set of specific tasks, and by decomposing each task into sub-tasks. We then created a **Surveillance Problem-Solving Ontology** in Protégé which automates each of the sub-tasks (Figure 3). With this ontology, our library becomes a computer-processable repository of problem-solving methods that can be indexed, queried and invoked when analyzing surveillance data.

As part of modeling syndromic surveillance in our problem-solving ontology, all analytic methods of our library are associated with a **method ontology** that defines the classes of data and knowledge on which a given method operates (Gennari et al. 1994). For example, methods in our library each have specific requirements for the structure and parameters of the statistical model on which they operate. Some operate at the population level, whereas others work at the individual level. Many algorithms expect time series data at varying granularities and certain algorithms require spatial data at several levels of aggregation. Within the surveillance problem-solving ontology, the method ontology makes explicit the data requirements of a problem-solving method, enabling that method to be applied uniformly to various data sources, independently of their storage format. The method ontology allows data sources to be mapped to appropriate problem solvers by reconciling the semantic differences between data and methods. Further, the method ontology facilitates the interoperation of analytic methods by identifying appropriate interactions between methods and different types of data. Overall, our framework provides a structure for incorporating surveillance algorithms into our system and for establishing the knowledge requirements of those methods. By making the characteristics of each method explicit, the ontology facilitates identifying methods suitable for a specific subtask in the overall task decomposition.

A Mediation Component for Integrating Data and Problem Solvers

Our data-source ontology provides a consistent mechanism for describing sources and elements of data in a way that allows them to be used concurrently by surveillance methods. However, surveillance methods operating on these data may have different input requirements. For example, different algorithms expect time series data at varying granularities and certain algorithms require spatial data at several levels of aggregation. Further, many methods used for surveillance analysis actually are generic, spatio-temporal methods that can operate on any kind of data, as long as the data have been construed in statistical terms that the methods expect. Keeping the input-output interface of these methods generic fosters their flexible reuse in different analytic strategies.

Because each problem solver in our library adheres to the declarative method ontology, our system explicitly specifies the types of data that each method can process (Figure 3). This explicit information about the data requirements of our methods, combined with the explicit information about the available sources of data from the data-source ontology, allowed us to devise a uniform mechanism to mediate data from multiple sources to various methods at run time. This mechanism involves two components, a **data broker** and a **mapping interpreter**. These components operate at the data level and at the ontological level, respectively, to reconcile the syntactic and semantic differences between incoming data and the input–output expectations of the methods. Together, the data broker and the mapping interpreter provide a semantic wall between analytic methods and raw data. Our approach allows us to make meaningful computations over disparate types of surveillance data without major reprogramming whenever BioSTORM incorporates a new data source or when we develop a new problem-solving method for the library.

A Data Broker for Integrating Data from Heterogeneous Sources

The data broker component uses the data-source ontology to allow problem solvers to read data transparently from many sources at run time. The component queries the data-source ontology for the description and context of a particular data source and constructs a stream of uniform data objects from raw data. First, the data broker accesses and retrieves data in the original location — currently relational databases or flat files — based on metadata describing the low-level data classes in the data-source ontology. Next, it formats the data and groups them as specified in the data-source ontology. The data broker packages the data with the appropriate context annotations to create data objects that are syntactically uniform in format and semantically unambiguous. The data objects are then ready to be sent to problem solvers that need to operate on them. This way, each problem solver receives a customized set of data objects and can ignore the raw data's original sources or formats.

A Mapping Interpreter for Unifying Data and Problem Solvers at the Ontological Level

Some problem solvers can operate directly on data supplied by the data broker. However, many surveillance methods in our library expect data in a format, conceptualization, or level of granularity that is different from the lower-level data objects provided by the data broker. In these cases, data must be supplied to the problem solvers in the appropriate representation, which requires mapping and transforming the data (Figure 4, top left).

We devised a **mapping ontology** (Gennari et al., 1994; Crubézy and Musen 2003) that enumerates the types of ontological transformations — or *mapping relations* — that enable data sources and data elements to match particular input–output specifications of different problem solvers (Figure 4, top right). For each data group in the data source ontology, specific mapping relations define the transformation of data elements into runtime inputs of problem solvers. These transformations range from simply renaming data-specific elements to the corresponding terms used by the method, to composing lexical or numerical expressions to match method terms as defined in the method ontology. For example, when configuring one of our surveillance methods to aggregate different data streams, where each stream reported on different 911 dispatches, we created a set of mappings to transform the contents of the different data streams into individual events as required by the aggregation method (Figure 4, bottom left).

After creating a knowledge base of data-to-method mapping relations, incoming data elements must be translated into a set of input instances for use by the particular method. Our **mapping interpreter** performs this task by processing the ontology-mapping relations for each data group and problem solver, and by generating streams of individual events that can be processed by the problem solvers. The mapping interpreter therefore transforms both

the contents and the semantics of the original data to conform to each problem solver's input requirements. The mapping ontology and mapping interpreter are tools for mediating knowledge and data from and to ontologies in Protégé (Sidebar 1). We successfully applied them within BioSTORM, in mapping SF 911 call record data to aggregation-method inputs (Figure 4, bottom right).

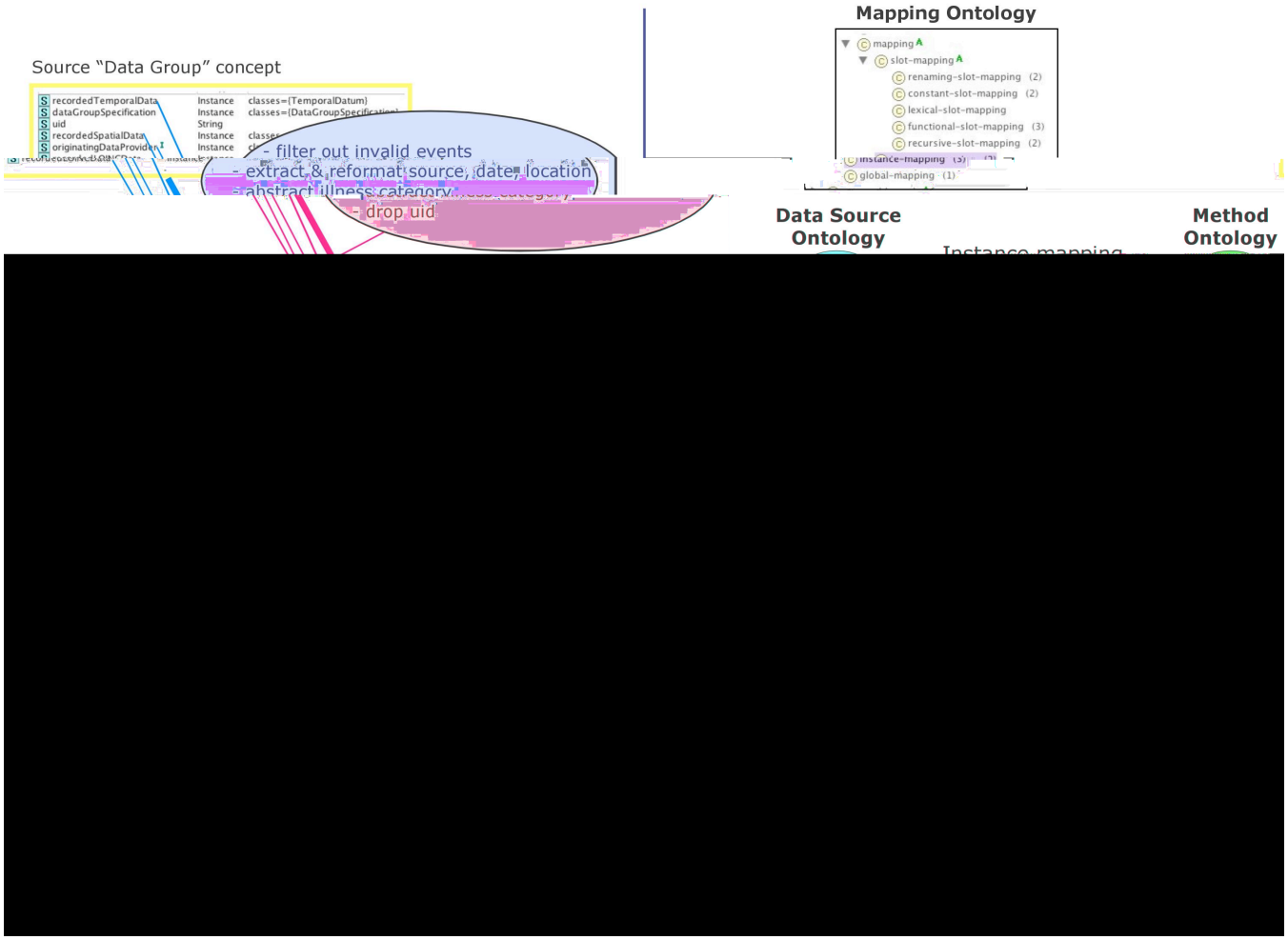

Figure 4. Mapping 911 call records Data Groups **to** Individual Events **for the** Aggregation **method.** (Top left) Various mismatches must be reconciled in order to process data groups as individual events required by the aggregation method. (Top right) Mapping relations specify transformations of instance data from the data-source ontology to the method ontology, both at the class and slot levels. A small, principled *mapping ontology* informs the process of creating such mapping relations. (Bottom left) A sample mapping relation, defined between the source's Data Group and the method's Individual Event highlights a slot-level relation specifying that the dataSource slot of the method's individual event should have the value "Dispatch911". (Bottom right) The result of interpreting the mapping relations includes the automatic generation of an Individual Event instance from a 911 Call Record Data Group instance.

A Control Structure for Deploying Surveillance Methods

As an overarching piece of the BioSTORM infrastructure, we developed RASTA, a control structure that coordinates data flow from raw representations to appropriate problem solvers, via the data broker and mapping interpreter (O'Connor et al., 2001). RASTA manages the unification of disparate data sources into semantically uniform data streams, maps them to multiple problem solvers, and deploys the problem solvers to conduct surveillance. RASTA uses the ontologies describing the surveillance data and the analytic methods to configure system components and monitor them, and ensures that the data broker passes the correct data at the correct time through the data mapper to the relevant problem solver. This entire process must be carried out efficiently, with potentially many data sources being sent to a variety of complex problem solver configurations operating in

parallel. Thus, the architecture does not store, access and manipulate data directly within the data-source and method ontologies; rather these ontologies hold abstract models and references to the various states of data.

RASTA uses a control system based on the JavaSpaces implementation of the Linda model to provide a distributed, knowledge-driven means for problem-solver deployment (http://www.sun.com/software/jini/specs/jini1.1html/js-spec.html). JavaSpaces provides a shared data store that allows Java processes to exchange data among eachother. We have developed a hybrid Linda/relational knowledge-driven model to provide efficient data flow in a deployed BioSTORM system. In this model, data are conceptually grouped into bundles based on shared semantic properties, which are described in terms of our data-source ontology. These bundles can be exchanged through JavaSpaces' data store. Data may be bundled spatially, temporally, or based on other semantic properties, such as all 911 calls for a particular ZIP code on a particular day. The raw data are stored in relational databases and semantic markers associated with the bundles reference these data. The data broker then controls the actual extraction of the bundled data from the appropriate relational database.

RASTA completely shields problem solvers from the details of these processes. Our approach allows BioSTORM to take advantage of the tremendous efficiencies afforded by the Linda model while offering a means for the system to scale up to handle large data sets. Additionally, our solution allows problem solvers to exchange data using high-level terms provided by the data-source ontology, freeing them from low-level data formatting concerns. When problem solvers require data in a form not provided by the data-source ontology, RASTA can invoke the mapping interpreter seamlessly to perform any necessary tailoring. The RASTA control structure thus provides a coherent, efficient runtime system that unifies data sources, knowledge bases, and problem solvers. It provides the basis for the BioSTORM computational system to support the modular, concurrent application, and the structured evaluation of multiple knowledge-driven analytic methods. Sidebar 2 shows the ontology-based deployment of a set of problem solvers over streams of 911 emergency data in San Francisco.

Discussion and Conclusion

BioSTORM demonstrates an end-to-end solution to many problems associated with data acquisition, integration, and analysis for public health surveillance. Its architecture builds solidly on long-standing work in AI concerning the use of ontologies for semantic integration, deployment of reusable problem-solving methods, mapping of problem-solving methods to domain ontologies, and parallelization methods for distributed problem solving. Most importantly, it utilizes the full potential of ontologies to represent all pieces of knowledge necessary in a system to acquire, access, integrate and mediate disparate data to judiciously chosen analytic methods. BioSTORM demonstrates how established AI methods can lead to rapid development of a robust approach to analyzing large volumes of disparate, noisy data.

Our generic approach is successful because ontologies offer a means to abstract details of data storage and analytic programming at the semantic level — where data and programs can interoperate meaningfully. Because analytic methods are independent of data sources and data mapping is isolated to methods at the semantic level, the approach enables a degree of flexibility not met by current surveillance systems. In that sense, our ontology-centered approach to syndromic surveillance provides a novel contribution to public health surveillance. Unlike special purpose solutions, our system can accommodate the unpredictible nature of data sources and outbreak patterns. When novel data sources are identified, developers can edit our data-source ontology and incorporate them in a straightforward manner. When new analytic methods are developed, they can be modeled in our library of surveillance problem solvers, and easily encoded and added to the RASTA control structure. Unlike existing systems for syndromic surveillance, BioSTORM does not require reprogramming whenever a new data source or a new analytic algorithm becomes available; instead, developers simply edit the associated ontologies — a task that is much more appropriate than maintaining low-level implementation details.

The ease with which our ontology-based approach accommodates changes to the system has implications that extend beyond system maintenance. The major difficulty with current deployed systems for syndromic surveillance is that not one of them has been rigorously evaluated. The homeland-security community has taken it on faith that these systems are useful (Bravata et al. 2004). However because syndromic surveillance of electronic data is a new discipline, identifying the optimal data and methods for detecting disease outbreaks will not be

possible until many combinations of data sources and analytic methods have been evaluated. BioSTORM's architecture offers a modular framework into which developers can drop new problem-solving methods and new data sources, and then measure system performance. Further, because of ontology-centered approach, the architecture provides the basis for correlating different data sources and selecting appropriate analytic methods at the ontological level, even when data and methods have different spatio-temporal granularities. We believe that BioSTORM may have immediate payoff in serving as a test bed on which to evaluate the relative contributions of surveillance data sources and analytics.

Areas of improvement to our approach include customizing the data-source ontology and the mediating components to account for emerging standards in biomedical data messaging such as CDC's PHINMS and HL7. Also, our next step in the classification of surveillance methods will be to define correspondences between specific disease agents and specific epidemic patterns. For example, influenza outbreaks tend to produce signals that are geographically diffuse, which rise to a peak incidence over weeks. Linking outbreaks to signal types would facilitate selection of analytic methods appropriate for detection of a specific outbreak. Finally, further testing of the performance of the BioSTORM control structure against custom-developed solutions would allow us to assess the computational efficiency of our ontology-driven implementation.

Public health agencies increasingly are using syndromic surveillance to monitor public health, and to detect bioterrorist attacks in particular. Rapid outbreak detection is important following a bioterrorist attack because public health interventions are generally more effective if applied early in the course of an outbreak as opposed to later. With an anthrax attack, for example, a delay in administering chemoprophylaxis of even hours can substantially lessen chances for survival. In this case, a system such as BioSTORM should sound an early warning by detecting an abnormal increase in clinic visits or pharmaceutical purchases before the first clinical diagnoses are made. A knowledge base of epidemic patterns associated with various disease agents would complete the system to help analytic methods determine the type of attack being detected.

Beyond syndromic surveillance, our architecture provides the conceptual and implementation components for developing many other data-source monitoring and analysis applications. Because all components of the architecture are generic, adapting the architecture for a new application, such as weather forecasting, is a matter of modeling application-specific knowledge elements into each ontology. Because our ontologies are all developed in the Protégé modeling environment, developers get significant tool support from the environment to perform those ontology extensions.

Acknowledgements

This work has been supported by the Defense Advanced Research Projects Agency, under a national research program for biosurveillance technology.

References

Bravata, D.M., McDonald, K.M. et al. (2004). A critical evaluation of existing surveillance systems for illnesses and syndromes potentially related to bioterrorism. Annals of Internal Medicine 140(11):910-922.

Buckeridge, D.L., Musen, M.A. et al. (2003). An analytic framework for space-time aberrancy detection in public health surveillance data. Proceedings of the American Medical Informatics Association Annual Symposium, Washington, D.C.: 120-4.

Buehler, J. W., Berkelman, R.L. et al. (2003). Syndromic surveillance and bioterrorism-related epidemics. Emerging Infectious Diseases 9(10): 1197-204.

Crubézy, M. and Musen, M.A. (2003). Ontologies in support of problem solving. Handbook on Ontologies. S. Staab and R. Studer, Springer-Verlag: 321-341.

Gennari, J.H., Tu, S.W., Rothenfluh, T.E. and Musen, M.A. (1994). Mapping domains to methods in support of reuse. International Journal of Human-Computer Studies 41: 399-424.

Lombardo, J., Burkom, H. et al. (2003). A systems overview of the Electronic Surveillance System for the Early Notification of Community-Based Epidemics (ESSENCE II). Journal of Urban Health 80(2 Suppl 1): i32-42.

O'Connor, M.J., Grosso, W.E., Tu, S.W., Musen, M.A. (2001). RASTA: a distributed temporal abstraction system to facilitate knowledge-driven monitoring of clinical databases. MedInfo2001, London, U.K., September, 2001.

Pavlin, J. A. (1999). Epidemiology of bioterrorism. Emerging Infectious Diseases 5(4): 528-30.

Pincus Z. and Musen, M.A. (2003) Contextualizing heterogeneous data for integration and inference. In: Proceedings of the American Medical Informatics Association Annual Symposium, Washington, D.C.: 514-8.

Tsui, F. C., Espino, J.U. et al. (2003). Technical description of RODS: a real-time public health surveillance system. Journal of the American Medical Informatics Association 10(5): 399-408.

Sidebar 1. Protégé: An ontology-based framework for building intelligent systems

Protégé is a methodology for building knowledge-based systems from three classes of reusable components (Gennari et al., 2003): (1) domain ontologies — models of the concepts in an application area and relations among those concepts (Gruber, 1993); (2) their associated knowledge bases containing domain facts; and (3) problem-solving methods — algorithms that apply generic patterns of reasoning to domain knowledge. In particular, we originally developed our methodology and associated tools for data and method integration using ontology-mapping techniques in the context of studying the reuse of problem-solving methods and domain ontologies.

The latest version of Protégé (http://protege.stanford.edu) is a free, open source ontology-development framework that provides a growing community of users with a suite of tools to construct domain models and knowledge-based applications. Protégé implements a knowledge model compatible with the Open Knowledge Base Connectivity protocol (Chaudhri et al., 1998), designed for interoperability among frame-based systems. In a frame-based modeling representation, an ontology consists of a set of *classes* organized in a subsumption hierarchy to represent the salient concepts of a domain; the properties associated to each of these concepts; and a set of *instances* of those classes — individual exemplars of the concepts that hold specific values for their properties. Protégé also supports other formats for representing knowledge bases, such as the Semantic-Web languages RDF (http://www.w3.org/RDF/) and OWL (http://www.w3.org/2001/sw/WebOnt/). Protégé provides both a wide set of user-interface elements for knowledge modeling and entry, and the capability to include custom-designed plug-in elements as extensions to an application (Musen et al., 2000; Noy et al., 2001). Protégé not only provides a robust, user-friendly environment for building ontologies, but also serves as a full-fledged server that can provide knowledge encoded in ontologies to any piece of program code invoking it.

In the BioSTORM system, Protégé provides a knowledge-based framework for our overall methodology as well as specific tools for knowledge-authoring and integration. In fact, most of the technology components that we built for BioSTORM rely on Protégé to supply and manage their ontologies. We developed custom knowledge-entry tools for annotating data sources within our data-source ontology, and for describing and organizing the analytical methods of our library according to our ontology for outbreak surveillance problem solving. Further, we developed a specific user-interface plug-in to Protégé to allow users to declare mapping relations between particular groups of data and the input specifications of particular problem solving methods (Crubézy et al., 2003).

References

Chaudhri, V. K., Farquhar, A. et al. (1998). OKBC: A programmatic foundation for knowledge base interoperability. AAAI'1998, Madison, Wisconsin, AAAI Press/The MIT Press.

Crubézy, M., Pincus, Z. and Musen, M.A. (2003). Mediating Knowledge between Application Components. In: Proceedings of the Semantic Integration Workshop of the Second International Semantic Web Conference (ISWC-03), Sanibel Island, Florida.

Gennari, J.H., Musen, M.A., Fergerson, R.W., Grosso, W.E., Crubézy, M., Eriksson, H., Noy, N.F. and Tu, S.W. (2003). The evolution of Protégé: an environment for knowledge-based systems development. International Journal of Human-Computer Studies 58(1): 89-123.

Gruber, T.R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition 5: 199-220.

Musen, M.A., Fergerson, R.W., Grosso, W.E., Noy, N.F., Crubézy, M. and Gennari, J.H. (2000). Component-based support for building knowledge-acquisition systems. Conference on Intelligent Information Processing (IIP 2000) of the International Federation for Information Processing World Computer Congress (WCC 2000), Beijing.

Noy, N.F., Sintek, M. et al. (2001). Creating and acquiring semantic web contents with Protégé-2000. IEEE Intelligent Systems 16(2): 60-71.

Sidebar 2. Surveillance of San Francisco 911 Emergency Dispatch Data

BioSTORM successfully deployed a set of problem solvers over streams of 911 emergency data in San Francisco (obtained for the year of 1999), based on the ontologies that form the backbone of the system (see Figure 5). The three Aberrancy Methods allow examination of the same data from different perspectives. In order to reduce the false positive rate, we first examined the results from the Likelihood Calculator which makes the fewest comparisons by collapsing input values across locations. The threshold for each Aberrancy Method was set at a level that declared an epidemic on average once a month, or 3.3% of the days in the training period. During the months of November and December of 1999, the Likelihood Calculator triggered an alert on 8 days for respiratory calls, 4 days for cardiovascular calls, and 2 days for trauma calls. As an example of how different methods can be used to confirm and localize aberrancies, we examined the results of the other two Aberrancy Methods for the day of the first respiratory alert - November 27, 1999. On that day, which coincides with a local peak in influenza admissions, the Minimum Aberrancy Calculator declared a respiratory alert for a ZIP code (94133) in the North East corner of the Peninsula. Examination of the output from the Poisson Aberrancy Calculator for that ZIP code reveals that on average, 0.26 respiratory calls were expected, and 2 calls were made.

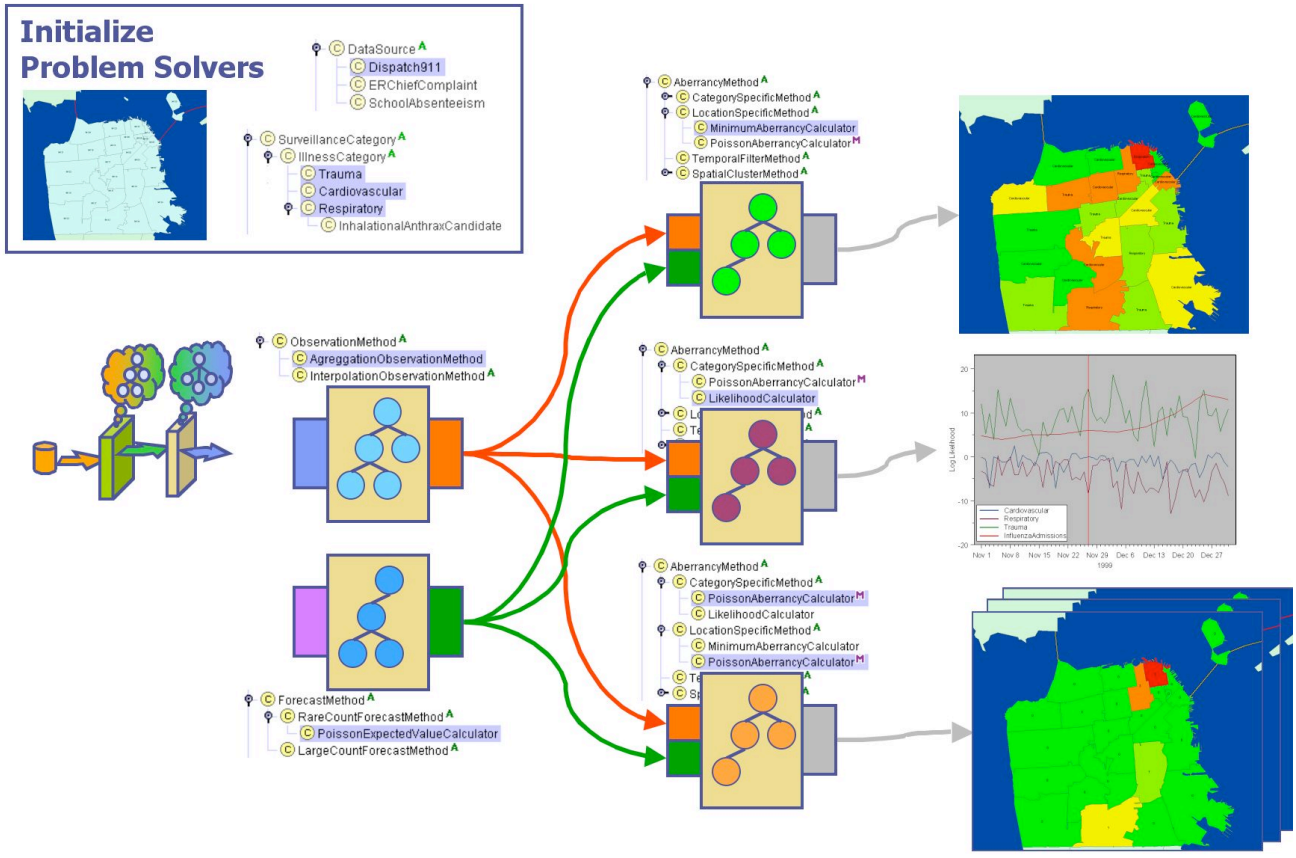

Figure 5. Deployment of three concurrent surveillance strategies on 911 data in San Francisco. Each strategy performs a different analysis on the same data to draw a different set of conclusions. The two leftmost methods convert raw data to aggregated, observed, and expected counts for each day. Three different analyses are then performed. The top rightmost method examines all syndrome counts by location and identifies the most unusual

syndrome count and p-values for each ZIP code, with the goal of finding spatial clusters of unusual counts that may span syndromes. The middle method examines the count for each syndrome separately, with counts aggregated across all ZIP codes — an approach that is useful for identifying outbreaks that are dispersed across a wide area. Finally, the bottom method examines the counts for each syndrome separately by spatial location, with the aim of identifying spatial clusters confined to one syndrome. All methods are initialized, connected and orchestrated by RASTA, based on our ontology of surveillance problem solvers and the associated input–output method ontology.