

Removing a Barrier to Computer-Based Outbreak and Disease Surveillance — The RODS Open Source Project

Jeremy U. Espino, M. Wagner, C. Szczepaniak, F-C. Tsui, H. Su, R. Olszewski, Z. Liu, W. Chapman, X. Zeng, L. Ma, Z. Lu, J. Dara
University of Pittsburgh, Pittsburgh, Pennsylvania

Corresponding author: Jeremy U. Espino, Real-Time Outbreak and Disease Surveillance Laboratory, University of Pittsburgh, Suite 500, Cellomics Building, 500 Technology Dr., Pittsburgh, PA 15219. Telephone: 412-383-8130; Fax: 412-383-8135; E-mail: jue@cbmi.pitt.edu.

Abstract

Introduction: Computer-based outbreak and disease surveillance requires high-quality software that is well-supported and affordable. Developing software in an open-source framework, which entails free distribution and use of software and continuous, community-based software development, can produce software with such characteristics, and can do so rapidly.

Objectives: The objective of the Real-Time Outbreak and Disease Surveillance (RODS) Open Source Project is to accelerate the deployment of computer-based outbreak and disease surveillance systems by writing software and catalyzing the formation of a community of users, developers, consultants, and scientists who support its use.

Methods: The University of Pittsburgh seeded the Open Source Project by releasing the RODS software under the GNU General Public License. An infrastructure was created, consisting of a website, mailing lists for developers and users, designated software developers, and shared code-development tools. These resources are intended to encourage growth of the Open Source Project community. Progress is measured by assessing website usage, number of software downloads, number of inquiries, number of system deployments, and number of new features or modules added to the code base.

Results: During September–November 2003, users generated 5,370 page views of the project website, 59 software downloads, 20 inquiries, one new deployment, and addition of four features.

Conclusions: Thus far, health departments and companies have been more interested in using the software as is than in customizing or developing new features. The RODS laboratory anticipates that after initial installation has been completed, health departments and companies will begin to customize the software and contribute their enhancements to the public code base.

Introduction

In October 1999, researchers at the University of Pittsburgh began developing the Real-Time Outbreak and Disease Surveillance system (RODS), with the goal of improving public health agencies' capability to detect a specific threat: a large-scale, surreptitious release of *Bacillus anthracis*. The rate of this technology's adoption, although accelerating, is not commensurate with the severity of the health threats posed by biologic terrorism, emerging infections, and common disease outbreaks. Such threats warrant rapid deployment; therefore, barriers to the technology's adoption need to be identified and removed.

This paper describes the evolution of the RODS system, previous efforts to transition the technology, and the rationale behind the creation of an open-source project. It also describes how the software is licensed, the infrastructure created to enable growth of the RODS open-source community, efforts to publicize the project, metrics collected to assess its progress, the software architecture of the latest version of RODS, and plans for additional software development.

RODS System Description

The first version of RODS collected patient chief-complaint data from eight hospitals in a single health-care system via Health Level 7 (HL7) (1) messages in real time, categorized these data into syndrome categories by using a classifier based on *International Classification of Diseases, Ninth Revision* (ICD-9) codes, aggregated the data into daily syndrome counts, and analyzed the data for anomalies possibly indicative of disease outbreaks. The system provided an Internet-based interface enabling users to view the data in graphs and maps (Figure 1). After demonstrating the feasibility of such a system within a single health-care system in Pittsburgh and conducting research to support the hypothesis that such a system could detect disease outbreaks (2,3), RODS' developers expanded the system to collect additional data types and then deployed RODS in multiple states. The application service provider (ASP) version of RODS at the University of Pittsburgh collects de-identified chief complaints from 76 hospitals in Pennsylvania, Utah, and Ohio (4,5) and also serves as the user interface for the National Retail Data Monitor

FIGURE 1. Sample time-series graphs* of syndromic surveillance data as displayed on the Epiplot user interface of the Real-Time Outbreak and Disease Surveillance (RODS) system



* In this example, graphs of pediatric gastrointestinal emergency department visits are shown alongside graphs depicting unit sales of antidiarrheals and electrolytes.

(NRDM), which collects and analyzes daily sales data for over-the-counter (OTC) medication sales (6,7).

The feasibility of rapid deployment of RODS was demonstrated during the 2002 Winter Olympics in Salt Lake City, Utah (4,8,9). In addition, the capability to integrate other surveillance data types (e.g., electronic laboratory reports [10], free-text chief complaints [11,12], laboratory orders, dictated radiology reports, dictated hospital reports [13–15], and poison control center calls [16]) was added. Much of the code (originally in Perl and C) was rewritten in Java,TM and basic research was conducted on data and algorithms relevant to this emerging science (17).

Technology Transition

The initial effort to make RODS software available involved licensing it for noncommercial use. In December 2002, the

University of Pittsburgh began offering the RODS system as compiled byte code, free of charge to public health departments. To date, >180 downloads of this version of the RODS system and >200 downloads of the Bayesian parser have been counted. Despite reports of successful installations in Hong Kong [David Wong, Hong Kong RODS Team, personal communication, May 15, 2003] and Missouri [Terry Tabor, Missouri Department of Health and Senior Services, personal communication, January 28, 2003], certain state health departments expressed interest in accessing the RODS source code.

Giving the software away without providing technical support soon proved insufficient. Using the RODS software requires expertise in database, network, geographic information system (GIS), HL7, and system management, capabilities not widely available at that time. Users made multiple requests for customization, support, and assistance with

installations, for which resources were not available. Therefore, in September 2003, the University of Pittsburgh released the RODS software under an open-source license, thereby creating the RODS Open Source Project to catalyze the sharing of knowledge and skills related to the software, including its design, installation, configuration, and customization.

Materials and Methods

This section describes the RODS Open Source Project, including the particular license under which RODS is distributed, the infrastructure created to enable growth of the RODS open-source community, methods for publicizing the project and recruiting developers, and the metrics collected to assess its progress.

GNU General Public License

RODS is distributed as open-source software under the GNU General Public License (GPL) (17), the same open-source license under which Linux[®] is distributed (18). Unlike the license under which RODS was initially released in December 2002, GPL permits anyone to use, copy, and modify RODS freely. GPL allows consultants and companies to use, install, support, and customize RODS and permits these entities to redistribute their enhanced versions of RODS, provided they make the source code available. This requirement fosters continuous software improvement, benefiting all users and preventing companies from creating proprietary, closed-source versions of RODS.

Support for Developers and Users

To coordinate community-based development of the code, the RODS Laboratory organized the Open Source Project. The RODS modules were classified into six functional areas: data collection, syndrome classification, data warehousing, database encapsulation, outbreak detection, and user interface. Specialists from the laboratory's research and development group named development leaders for each functional area. These development leaders are responsible for recommending new features based on user requests and evaluating whether a developer has the qualifications to contribute source code.

Online resources were created to support the Open Source Project, including the RODS Laboratory website (<http://www.health.pitt.edu/rods>) and a project website hosted on Sourceforge[™] (<http://openrods.sourceforge.net>). The latter site provides standard software project management tools (a concurrent versions system server and patch submission area enabling developers to contribute code), e-mail lists

enabling developers and users to communicate, a software-bug reporting system, contact information for the development leaders, and source code for stable versions of the system.

Recruitment of Developers and Users

E-mail announcements were sent to 181 persons who had previously downloaded the byte-compiled releases and to all 226 users in the United States who held passwords to the RODS ASP system. Users were given an opportunity for a face-to-face meeting with the core developers at two national conferences, the 2003 National Syndromic Surveillance Conference in New York City and the 2003 American Medical Informatics Fall Symposium in Washington, D.C. Project leaders of other computer-based surveillance projects were also invited.

Metrics

The following metrics are collected monthly to manage the project and assess its progress:

- cumulative number of installations;
- cumulative number of developers who have contributed code;
- number of new features;
- funding sources;
- cumulative number of mailing list subscribers (one general mailing list, one for announcements, and one for development questions);
- total website page views;
- total downloads of source code;
- number of e-mail announcements sent;
- cumulative number of inquiries from consultants and companies;
- cumulative number of inquiries from health departments;
- cumulative number of inquiries from academics; and
- cumulative number of inquiries from other groups.

The number of installations and the number of contributing developers are considered the two most important metrics.

Results

Current Software Architecture of RODS Version 2.0 and Features in Development

A complete technical description of RODS has been published (8). This section describes the system's software architecture and how the modules that comprise that architecture can be used to accomplish different surveillance tasks.

RODS 2.0 consists of >42,000 lines of Java code contributed by a team of eight programmers. RODS is a modular system that adheres to CDC's National Electronic Disease Surveillance System (NEDSS) (19) and Public Health Information Network (PHIN) (20) standards so that any of the components can be incorporated into a foreign surveillance system or used to create a native end-to-end RODS system.

The RODS software architecture consists of six functional areas: data collection, syndrome classification, data warehousing, database encapsulation, outbreak detection, and user interface (Figure 2). Within the following categories, additional modules are being developed under the Open Source Project (Table 1):

- **Data collection.** The data-collection modules consist of 1) an HL7 listener that accepts and maintains connections from a hospital's HL7-integration engine; 2) an HL7 parser that extracts patient-visit data from HL7 messages; and 3) a text-file parser that extracts patient-visit data from text files uploaded in batches by non-HL7-capable hospitals. In addition to modules to parse patient data from HL7 messages, modules are being developed to parse microbiology culture results from HL7 messages and to import poison center call data to RODS.

Another module is proposed that will fully integrate detailed OTC medication sales data from the NRDM. Also planned is an extensible markup language (XML) module that works with proposed or currently used XML-document-type definitions for public health surveillance data (21,22).

- **Syndrome classification.** RODS Version 2.0 consists of a single module for syndrome classification, Complaint Classifier (CoCo) (12). CoCo uses a naïve Bayesian classifier to assign a free-text chief complaint to a syndrome category. These syndrome categories are user-specifiable, and the mappings are created automatically through machine learning from a user-provided training set.

The RODS Laboratory has rewritten (in Java) and intends to release a module for ICD-9-based classification (8). Additional classification modules, including keyword-based methods and additional natural language processing modules to identify radiology reports indicative of inhalational anthrax (15), are in development.

- **Data warehousing.** These modules function to store and provide efficient access to surveillance data. RODS efficiently stores and retrieves time-series data from the database through a data warehouse. The data-warehousing module consists of a cache table updater that keeps running counts of the number of visits for each syndrome, stratified by age and sex.

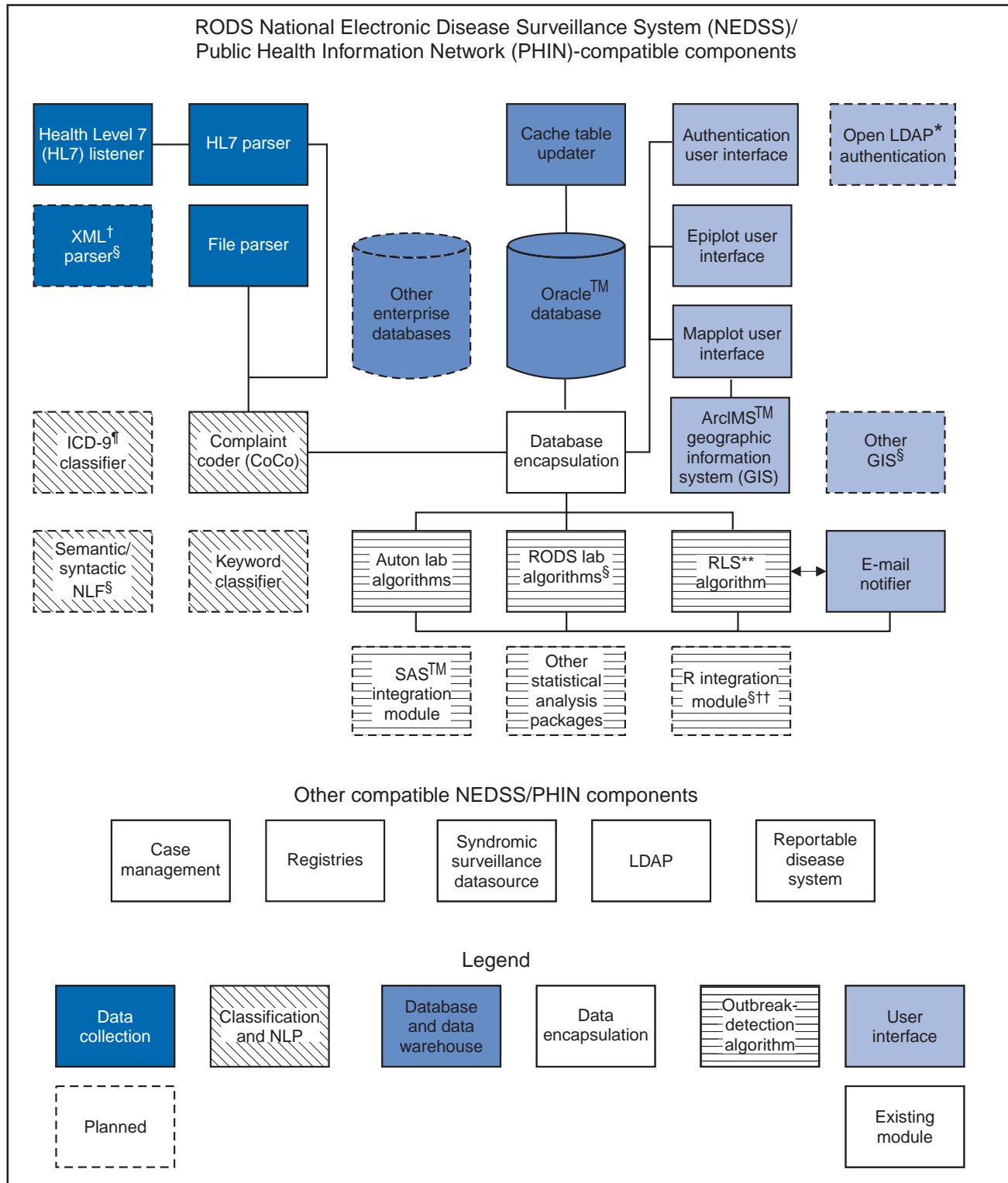
RODS 2.0 assumes the existence of an Oracle™ database. However, RODS does not use Oracle-specific structured query language (SQL) functions (e.g., database triggers), and a port to an alternative relational database system (e.g., PostgreSQL or Microsoft SQL Server™) should be straightforward.

- **Database encapsulation.** The database-encapsulation modules, written as Enterprise Java Beans™ (EJBs), function to retrieve preprocessed time-series data and case details (e.g., the patient's free-text chief complaint) from the database. In Java, EJBs provide a framework for creating readily accessed software objects that incorporate standard methods for security, database access, transactions, scalability, and communication. The EJBs shield developers from the database schema and standardize how the surrounding modules (e.g., the user interface modules) access the database.
- **Detection algorithm.** The detection-algorithm modules provided in the current open-source release include an implementation of the recursive least-squared (RLS) algorithm (23) and an initial implementation of a wavelet-detection algorithm. The RLS algorithm can detect sudden increases in daily surveillance data counts (e.g., an increase in the number of respiratory-type visits that would accompany a large-scale, covert release of *Bacillus anthracis*). The wavelet algorithm can automatically model weekly, monthly, and seasonal data fluctuations. NRDM uses wavelet modeling to indicate zip-code areas in which OTC medication sales are substantially increased; this algorithm will be applied to the analysis of health-care registration data.

Another set of modules are planned that will enable any outbreak-detection algorithm to analyze data from the system. Currently, the architecture allows algorithms written or wrapped in Java to retrieve data directly from the database-encapsulation modules. A module will be released that outputs data as common text files so that stand-alone algorithms and statistical software packages can be used to analyze the data. This method was used by the What's Strange About Recent Events algorithm (WSARE) to analyze data from RODS during the Salt Lake 2002 Olympic Winter Games (24).

- **User interfaces.** These modules 1) authenticate users, 2) display surveillance data as time-series graphs, and 3) work with a GIS to depict data spatially. The graphing and GIS modules consist of Java server pages and servlets that use JFreeChart, an open-source graphing package, and the GIS functions of Environmental Systems Research Institute's ArcIMS™ software.

FIGURE 2. Software architecture of the Real-Time Outbreak and Disease Surveillance (RODS) system



* LDAP = Lightweight directory access protocol.
 † XML = extensible markup language.
 § Has General Public License-compatible open-source components.
 ¶ ICD-9 = *International Classification of Diseases, Ninth Revision*.
 ** RLS = Recursive least-squared.
 †† An open-source statistical computing software program.

TABLE 1. Existing features of the Real-Time Outbreak Disease Surveillance (RODS) system, version 2.0, and features awaiting development

RODS feature	Exists in RODS 2.0	Exists as GPL*-compatible source code	Needs to be developed or tested
Data-collection modules			
Health Level 7 (HL7) listener	X		
HL7 parser for microbiology reports			X
HL7 parser for admissions, discharge, and transfer (ADT) messages	X		
Text file parser	X		
XML parser		X	
Syndrome-classification modules			
Simple Bayesian syndrome classifier	X		
Syntactic/semantic natural language processing (NLP) classifier		X	
Keyword classifier			X
ICD-9 [†] classifier			X
Multiple data-type classifier		X	
Data-warehousing modules			
Diverse database options		X	
Integrated data-warehouse engine			
Aggregation by sex and age	X		
Outbreak-detection modules			
Integrates with external statistical analysis tools	X	X	X
Recursive least-squared (RLS) detection algorithm	X		
User-interface modules			
Manual data-entry interface		X	
Diverse geographic information system software options		X	
Lightweight directory access protocol (LDAP) interface		X	
Time-series graphing	X	X	
Options and preferences	X		
Custom jurisdictions	X		
E-mail notifier	X		
Database-encapsulation modules			
Database encapsulation	X		

* GNU General Public License.

[†] *International Classification of Diseases, Ninth Revision.*

Certain state health departments have requested Lightweight Directory Access Protocol (LDAP) support to enable the creation of seamless links between existing state surveillance systems and the surveillance functions provided by RODS; outside development of such a module is encouraged.

State, local, or national health departments can use RODS modules to collect, analyze, and view hospital surveillance data and to view OTC medication sales data from NRDM. A health department can use a subset of these modules to accomplish a specific surveillance task (e.g., receiving and processing free-text chief complaints from hospitals), or it can use all of them (with the RODS database, analytic modules, and user interface) to create an end-to-end surveillance solution. (Examples of how health departments can mix and match RODS modules for different surveillance tasks are available at <http://openrods.sourceforge.net>.)

Project Metrics

A total of 480 e-mail announcements about the RODS Open Source Project were sent during the first 3 months of the project. This publicity generated 5,370 page views of the project website, 59 downloads of the source code, and 14 new members to the project mailing lists. One additional installation is using the open-source version of RODS.

To date, users are more interested in using the software “as is” and less interested in collaborative feature development. For example, users have asked when the ICD-9 classifier module will be released or whether the system yet works with Microsoft SQL Server.[™] Developers at the RODS Laboratory contributed four new features (drilldown of age and sex, customized jurisdictions, a simplified GIS interface, and user preferences) (Table 2). However, at least one health department and one consulting company have expressed interest in collaborating to develop a module that will import XML data into RODS.

TABLE 2. Monthly metrics for the Real-Time Outbreak Disease Surveillance (RODS) Open Source Project — September–November 2003

Metric	September	October	November
Number of e-mail announcements sent out	406	0	74
Total page views on website	1,968	1,764	1,638
Total downloads of source code	18	19	22
Cumulative number of members on site mailing lists	7	8	14
Cumulative number of installations	3	4	4
Cumulative number of inquiries from consultants and companies	7	8	9
Cumulative number of inquiries from health departments	5	6	7
Cumulative number of inquiries from academics	1	1	1
Cumulative number of inquiries from other groups	2	3	3
Cumulative number of developers	8	8	8
Number of new features	0	0	4
Funding sources	1	1	1

Discussion

The goal of the RODS Open Source Project is to accelerate the deployment of computer-based outbreak and disease surveillance systems by writing high-quality surveillance software and catalyzing the formation of a community of users, developers, consultants, and scientists. In the initial years of computer-based outbreak and disease surveillance system development, the main barriers to deployment appeared to be doubts about its efficacy, cost of the technology, concerns about the cost and effect of false alerts on the practice of public health, and legal and administrative issues (25,26). Basic research about data and detectability has been conducted to address concerns about efficacy (2,3,27–29). To address concerns about the effects of false alerts, the RODS laboratory has deployed systems and discovered that persons working in health departments could incorporate the output of these systems into their workflows (4,7). The deployments also established that the cost and effort of deployment is much lower than expected. Finally, the deployments demonstrated that certain concerns about privacy could be addressed. The Health Information Portability and Accountability Act of 1996 (HIPAA), which had not yet become law, nevertheless had a substantial inhibitory effect on hospitals and other covered entities that had data needed by the project. The enactment of the final privacy rule, precedents set by system deployments (4,30–32), and new state laws have helped address certain concerns of data providers (33).

Open-source projects can create a community of like-minded persons — scientists, programmers, consultants, and users — who have the vision of creating innovative, well-

supported software. The importance of catalyzing such a community cannot be overstated. It can strengthen the position of information technology (IT) managers and public health officials who wish to deploy computer-based surveillance systems during planning deliberations. They will be able to assure their supervisors that source code is available, that a pool of developers and consultants exists who can be hired to support the health department if needed, and that ongoing projects in other health departments can help them predict project costs and set appropriate timelines.

The RODS Open Source Project enables public health professionals to have a greater role in developing IT solutions to the problem of early detection. Just as public health researchers publish their results in scientific journals, so can they contribute publicly available IT solutions to the

RODS Open Source Project. This role might become more apparent as public health personnel become increasingly knowledgeable about public health informatics and work more closely with IT subcontractors and consultants.

Continued goals for the RODS Open Source Project are to increase the number of deployments, developers, and supporters of the software. The proposed path for RODS software development is to increase the number of data types the system can accept and implement a range of high-performance outbreak-detection algorithms. One consulting company and one health department have separately expressed interest in collaboratively developing an XML module that can parse non-RODS data sources. The RODS Laboratory and its collaborators at the Auton Laboratory will continue to develop outbreak-detection algorithms (e.g., the wavelet-detection module and WSARE, respectively).

Conclusion

The RODS Open Source Project is making software modules available that span the spectrum of processing tasks involved in public health surveillance. Through open source, the project hopes to accelerate the deployment of real-time public health surveillance by lowering costs, increasing reliability, preventing vendor lock-in, and ensuring software customizability. By catalyzing the formation of a community of open-source public health surveillance software advocates, this approach will result in a high-quality software product that achieves mainstream acceptance.

Acknowledgments

The RODS Open Source Project is supported by the Pennsylvania Department of Health Bioinformatics Grant ME-107.

References

- Health Level Seven, Inc. Health Level Seven. Ann Arbor, MI: Health Level Seven, Inc., 2001. Available at <http://www.hl7.org>.
- Espino JU, Wagner MM. The accuracy of ICD-9 coded chief complaints for detection of acute respiratory illness. *Proc AMIA Symp* 2001;164–8.
- Tsui F-C, Wagner MM, Dato V, Chang CC. Value of ICD-9 coded chief complaints for detection of epidemics. *Proc AMIA Symp* 2001;711–5.
- Gesteland PH, Gardner RM, Tsui F-C, et al. Automated syndromic surveillance for the 2002 Winter Olympics. *J Am Med Inform Assoc* 2003;10:547–54.
- Tsui F-C, Espino JU, Dato VM, Gesteland PH, Hutman J, Wagner MM. Technical description of RODS: a real-time public health surveillance system. *J Am Med Inform Assoc* 2003;10:399–408.
- Wagner MM, Robinson J, Tsui F-C, Espino JU, Hogan W. Design of a national retail data monitor for public health surveillance. *J Am Med Inform Assoc* 2003;10:409–18.
- Wagner MM, Espino J, Hersh J, et al. National retail data monitor for public health surveillance. *MMWR* 2004;53(Suppl):40–2.
- Tsui F-C, Espino JU, Wagner MM, et al. Data, network, and application: technical description of the Utah RODS Winter Olympic Biosurveillance System. *Proc AMIA Symp* 2002;815–9.
- Gesteland PH, Wagner MM, Chapman WW, et al. Rapid deployment of an electronic disease surveillance system in the state of Utah for the 2002 Olympic Winter games. *Proc AMIA Symp* 2002;285–9.
- Panackal AA, M'ikanatha NM, Tsui F-C, et al. Automatic electronic laboratory-based reporting of notifiable infectious diseases. *Emerg Infect Dis* 2001;8:685–91.
- Chapman WW, Christensen LM, Wagner MM, et al. Classifying free-text triage chief complaints into syndromic categories with natural language processing. CBMI report series. Pittsburgh, PA: University of Pittsburgh, Center for Biomedical Informatics, 2002.
- Olaszewski R. Bayesian classification of triage diagnoses for the early detection of epidemics. In: Russell I, Haller S, eds. *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference*. Menlo Park, CA: AAAI, 2003;412–7.
- Chapman WW, Dowling JN, Wagner MM. Fever detection from free-text clinical records for biosurveillance. *J Biomed Inform* 2004;37:120–7.
- Chapman WW, Espino JU, Dowling JN, Wagner MM. Detection of acute lower respiratory syndrome from chief complaints and ICD-9 codes. CBMI report series. Pittsburgh, PA: University of Pittsburgh, Center for Biomedical Informatics, 2003.
- Chapman WW, Wagner M, Cooper G, Hanbury P, Chapman B, Harrison L. Creating a text classifier to detect chest radiograph reports consistent with features of inhalational anthrax. *J Am Med Inform Assoc* 2003;10:494–503.
- Zeng X, Wagner MM. Accuracy, speed, and completeness of data collection by poison control center nurses for the investigation of outbreaks of acute diarrhea. CBMI report series. Pittsburgh, PA: University of Pittsburgh, Center for Biomedical Informatics, 2003.
- Free Software Foundation, Inc. GNU general public license. Boston, MA: Free Software Foundation, Inc., 1991. Available at <http://www.gnu.org/licenses/gpl.html>.
- Raymond ES. *The cathedral and the bazaar: musings on Linux and Open Source by an accidental revolutionary*. Rev. ed. Beijing: Cambridge, MA: O'Reilly, 2001.
- CDC. National Electronic Disease Surveillance System: the surveillance and monitoring component of the Public Health Information Network. Atlanta, GA: US Department of Health and Human Services, CDC, 2004. Available at <http://www.cdc.gov/nedss/>.
- CDC. Public Health Information Network. Atlanta, GA: US Department of Health and Human Services, CDC, 2003. Available at <http://www.cdc.gov/phin/>.
- Lober WB, Trigg LJ, Karras BT, et al. Syndromic surveillance using automated collection of computerized discharge diagnoses. *J Urban Health* 2003;80(2 Suppl 1):i97–106.
- Barthell EN, Cordell WH, Moorhead JC, et al. The Frontlines of Medicine Project: a proposal for the standardized communication of emergency department data for public health uses including syndromic surveillance for biological and chemical terrorism. *Ann Emerg Med* 2002;39:422–9.
- Hayes M. *Statistical digital signal processing and modeling*. New York, NY: John Wiley & Sons, Inc, 1996.
- Wong WK, Moore A, Cooper G, Wagner M. WSARE: What's Strange About Recent Events? *J Urban Health* 2003;80(2 Suppl 1):i66–75.
- Broome CV, Pinner RW, Sosin DM, Treadwell TA. On the threshold. *Am J Prev Med* 2002;23:229.
- Reingold A. If syndromic surveillance is the answer, what is the question. *Biosecur Bioterr* 2003;1:1–5.
- Hogan WR, Tsui F-C, Ivanov O, et al. Early detection of pediatric respiratory and diarrheal outbreaks from retail sales of electrolyte products. *J Am Med Inform Assoc* 2003;10:555–62.
- Ivanov O, Wagner MM, Chapman WW, Olaszewski RT. Accuracy of three classifiers of acute gastrointestinal syndrome for syndromic surveillance. *Proc AMIA Symp* 2002;345–9.
- CDC. Annotated bibliography for syndromic surveillance. Atlanta, GA: US Department of Health and Human Services, CDC, 2003. Available at <http://www.cdc.gov/epo/dphsi/syndromic/index.htm>.
- Moran GJ, Talan DA. Syndromic surveillance for bioterrorism following the attacks on the World Trade Center—New York City, 2001. *Ann Emerg Med* 2003;41:414–8.
- Lober WB, Karras B, Wagner MM, et al. Roundtable on bioterrorism detection: information system-based surveillance. *J Am Med Inform Assoc* 2002;9:105–15.
- Lewis M, Pavlin J, Mansfield J, et al. Disease outbreak detection system using syndromic data in the greater Washington, DC, area. *Am J Prev Med* 2002;23:180.
- Broome CV, Horton HH, Tress D, Lucido SJ, Koo D. Statutory basis for public health reporting beyond specific diseases. *J Urban Health* 2003;80(2 Suppl 1):i14–22.