User Centred Networked Health Care
A. Moen et al. (Eds.)
IOS Press, 2011

doi:10.3233/978-1-60750-806-9-160

Event-Driven Architecture for Health Event Detection from Multiple Sources

Kerstin DENECKE[a,1], Göran KIRCHNER[b], Peter DOLOG[c], Pavel SMRZ[d], Jens LINGE[e], Gerhard BACKFRIED[f], Johannes DREESMAN[g]
[a] *L3S Research Center, Hannover, Germany*
[b] *Robert Koch Institut, Berlin, Germany*
[c] *Aalborg University, Aalborg, Denmark*
[d] *Brno University of Technology, Brno, Czech Republic*
[e] *Joint Research Centre, Ispra, Italy*
[f] *SAIL Labs Technology, Vienna, Austria*
[g] *Niedersächsisches Landesgesundheitsamt, Hannover, Germany*

Abstract. Early detection of potential health threats is crucial for taking actions in time. It is unclear in which information source an event is reported first and, information from various sources can be complementing. Thus, it is important to search for information in a very broad range of sources. Furthermore, real-time processing is necessary to deal with the huge amounts of incoming data in time. Event-driven architectures are designed to address such challenges. This will be shown in this paper by presenting the architecture of a public health surveillance system that follows this style. Starting from concrete user requirements and scenarios, we introduce the architecture with its components for content collection, data analysis and integration. The system will allow for the monitoring of events in real-time as well as retrospectively.

Keywords. Epidemic Intelligence, Text Mining, Disease Surveillance, Event-driven architecture

1. Introduction

Various factors such as globalization, climate change, or behavioral changes contribute to continuous emergence of public health hazards. A health hazard can be described as a sudden, unexpected event, incident or circumstance, confronting public health officials with a situation threatening the health of people and society with substantial consequences, e.g., outbreak of an infectious disease like swine flu or measles. Early detection of disease activity followed by an appropriate assessment of its risk and a corresponding reaction can help reduce and manage risk produced by health hazards [1].

Surveillance systems aim at supporting health officials in obtaining information on potential health hazards as early as possible. A main requirement of such systems is the processing of incoming data in real time. Event-driven architectures are designed to support this kind of processing. It is an architectural style that orchestrates behavior around the production, detection and consumption of events [3]. An event in this context is some message, token, count or pattern that can be identified within an

[1] Corresponding author

ongoing stream of monitored inputs, such as network traffic, specific error conditions or signals, thresholds crossed, counts accumulated etc.

In this paper, we provide an overview on the characteristics of event-driven architectures for disease surveillance and present an architecture that follows this style. After providing an overview on related work in section 2, one contribution of this paper is the presentation of requirements for improved event-based surveillance systems. Then, the suggested event-driven system architecture for a disease surveillance system is described. The paper will finish with lessons learned from user feedback sessions and with conclusions on future work.

2. Related Work

Epidemic intelligence is the science of collecting, filtering, verifying and analyzing information related to potential health threats [1]. It includes traditional approaches, such as public health surveillance where data from hospitals and laboratories are monitored. Further, it comprises systems that scan the Internet for relevant events (e.g. news wires, media sources or websites, twitter) [9, 10]. For example, the Medical Information System (MedISys, [2]) is a fully automatic public health surveillance system to monitor reporting on human and animal infectious diseases, and other public health threats. The system retrieves news articles from the Internet and classifies them according to pre-defined multilingual categories. It identifies entities like organizations, persons and locations. Using Pattern-based Understanding and Learning System (PULS, http://puls.cs.helsinki.fi/medical, [2]) event information is extracted and clustered.

The main objective of event-driven architectures is to facilitate immediate information dissemination and reactive business process execution [11, 12]. In contrast to other architectural styles, they are characterized by actuality (events are monitored in real time), efficiency (huge amount of data is processed), robustness (components can be added and replaced easily) and their flexibility and adaptability (new types of events can be integrated easily) [3]. So far, mainly business applications took advantage from the event-driven style. As summarized by Li [4], other application domains require real-time processing, too, and could thus benefit from the same paradigm. In this paper, the focus is on the application domain of disease surveillance. Systems following the event-driven architectural style have not yet been described for this domain.

3. Requirements and Scenarios

In a workshop and discussions with representatives of health organizations like World Health Organization or European Center of Disease Control, we collected requirements for an improved system for disease surveillance. They concern four main issues:

1. **Content collection**: The system should monitor a *broad range of sources* in a multitude of languages and from being broadcasted or produced around the globe. Complementing results from existing systems need to be accessible through a single user interface. Besides monitoring diseases and their mentions, it is of interest to monitor symptoms and their mentions as well as behavioral changes.

2. **Result filtering:** Users don't want to be overwhelmed with information. Thus, results need to be carefully *filtered* according to various filter criteria such as relevance, novelty, source of information etc.
3. **Result presentation:** Event information should be presented in a *structured, appropriate and user-friendly way* and allow accessing the original information sources for event verification processes. The user would like to interact with results, e.g. by narrowing the result set or by redefining his interest.
4. **User feedback and interaction:** Interactions of interest include a) specification of essential signal and event information (e.g., disease name, location), b) selecting result presentation formats and storing event information and c) providing feedback for future result adaptations.

A potential use case is the *user notification scenario* where a system regularly provides information on new upcoming health threats identified in various information sources. The input is therefore a specified user interest (called signal definition). The system outputs signals matching this definition or those that might be of potential interest (see Fig. 1 for some example output).

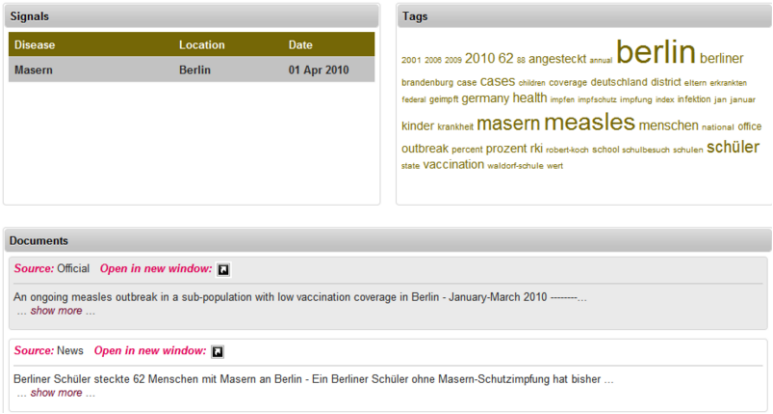

Figure 1. Screenshot of the result page: One signal has been generated and related information is given.

4. Architecture

Our architecture consists of various technical components that realize the individual processing steps. The components interact via web services. Collected (textual) content and processing results are stored in a database and transferred as RSS feeds. Figure 2 shows the information flow between the single components. For simplification reasons, the database accesses are not shown in the diagram. Knowing the user interest specified in the signal definition, the system continuously monitors the incoming text and data streams for relevant events. Once patterns of interest are identified, appropriate services for pattern analysis and interpretation are triggered and an alert is produced. The single components will be described in the following.

The *Content Collection Component* collects continuously data from various sources including TV, radio, online news, blogs and Twitter. TV and radio data is collected via satellite and transcribed by the SAIL Media Mining System (http://www.sail-technology.com/products/commercial-products/media-mining-

indexer.html*)*. Medical Blog data includes MedWorm (http://www.medworm.com/) listed blogs and manually selected blogs collected through corresponding APIs (e.g., Twitter, MedWorm). Data collected by other surveillance systems can be integrated easily when the data is made accessible via RSS (e.g., MedISys data [2]).

The *Document Analysis Component* filters and pre-processes the collected (textual) data before making it available for the event detection component. Pre-processing includes filtering of irrelevant data, recognition of mentions of disease names and symptoms, locations, time etc. The latter is realized by OpenCalais (http://www.opencalais.com). The documents are analysed linguistically by Minipar (http://webdocs.cs.ualberta.ca/~lindek/minipar.htm). As a result, a set of documents annotated with named entities and linguistic structures is produced. These tagged documents are indexed and made available through MG4J (Managing Gigabytes for Java, http://mg4j.dsi.unimi.it/) which is a free full-text search engine for large document sets.

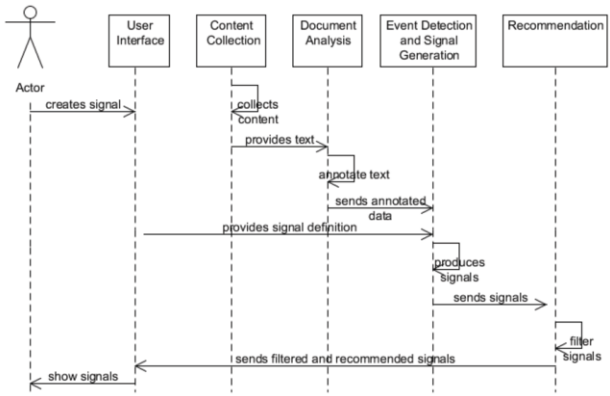

Figure 2. Information flow

The *Event Detection and Signal Generation Component* exploits the tagged documents to identify patterns of interest and to produce signals. It works in two modes: The *unsupervised event detection* (introduced in [6]) groups documents into clusters by a retrospective event detection algorithm and intends to identify signals that might be of potential user interest, not particularly matching the signal definition. These clusters are interpreted as signals and exploited by the recommendation component (see below). The *supervised event detection* considers the signal definition entered by the user. Given this information, it first retrieves data from the data repository that is relevant for the specified information need. The system then identifies segments (e.g., sentences, paragraphs) in the relevant documents by means of a supervised machine learning algorithm (see [8] for details). This information is then exploited by standard statistical algorithms for biosurveillance (e.g. CUSUM, Farrington [7]) to produce signals as alerts for health officials (*signal generation)*.

The *Recommendation Component* gets as input the document clusters or the calculated signals and either selects those that are of interest for the user according to his profile or ranks the signals appropriately. This component requires the user profile that consists of information on a specified signal definition as well as user feedback from previous searches and user interactions. The ranking of signals in the result

presentation is adapted to the user interest and irrelevant signals can be filtered out. The produced signals are presented in the user interface.

The *user interface* allows a user to specify his interest in terms of a signal definition. It collects information on disease names or symptoms, locations to be considered by the surveillance system. Further, the generated signals and related information on indicators and the information source are presented to the user. Users are enabled to browse through the results. Various visualization methods are applied to present the results in an easy understandable way (e.g., as word clouds, in maps or graphs).

5. Conclusion

In this paper, the architecture for an event-driven disease surveillance system has been introduced. The system allows monitoring a broad range of sources including indicator data from traditional surveillance. The first versions of the single components are currently integrated and will be tested by epidemiologists with real world data in near future. From the user feedback we learned so far, that it is necessary to carefully select the social media sources in order not to get to many false alarms. In future work we will focus on testing and improving the algorithms.

Acknowledgements: This research is part of the M-Eco project funded partly under 247829 by the European Commission.

References

[1] Paquet C, Coulombier D, Kaier R, Ciotti M. Epidemic intelligence: a new framework for strengthening disease surveillance in Europe. *Euro Surveill. 2006*; 11(12)

[2] Steinberger R, Fuart F, van der Goot E, Best C, von Etter P, Yangarber R. Text Mining from the Web for Medical Intelligence. In: Perrotta D, Piskorski J, Soulié-Fogelman F, Steinberger R, eds.: *Mining Massive Data Sets for Security*, OIS Press. (2008) The Netherlands

[3] Bruns R, Dunkel J. *Event-Driven Architecture: Softwarearchitektur für ereignisgesteuerte Geschäftsprozesse.* Springer, Berlin; 1st Edition. (29. Mai 2010)

[4] Li C-S. Real-time event driven architecture for activity monitoring and early warning. Emerging Information Technology Conference, 2005.

[5] Faensen D, Claus H, Benzler J, et al.: SurvNet@RKI – a multistate electronic reporting system for communicable diseases. *Euro Surveill 2006*;11(4):100-3

[6] Fisichella M, Stewart A, Denecke K, Nejdl W. Unsupervised Public Health Event Detection for Epidemic Intelligence. CIKM'10, October 25-29, 2010, Toronto, Ontario, Canada

[7] Höhle M. surveillance: An R package for the surveillance of infectious diseases, *Computational Statistics (2007),* 22(4), pp. 571-582

[8] Stewart A, Denecke K. Using ProMED Mail and MedWorm Blogs for Cross-Domain Pattern Analysis in Epidemic Intelligence. In: Safran C, Reti S, Marin HF, eds. Studies in Health Technology and Informatics: MEDINFO 2010, IOS Press, Amsterdam, 2010, pp. 473-481

[9] Corley CD, Cook DJ, Mikler AR, Singh KP. Using Web and Social Media for Influenza Surveillance. Book chapter in *Advances in Computational Biology*, Springer, 2010

[10] Linge JP, Steinberger R, Weber TP, et al.: Internet surveillance systems for early alerting of health threats. Editorial. *Eurosurveillance*, Volume 14, Issue 13, 02 April 2009

[11] Michelson BM. *Event-Driven Architecture Overview*, Patricia Seybold Group, February 2, 2006

[12] Chandy KM. Event-Driven Applications: Costs, Benefits and Design Approaches, *California Institute of Technology*, 2006