

Implementing a Real-time Complex Event Stream Processing System to Help Identify Potential Participants in Clinical and Translational Research Studies

Susan Weber PhD¹, Henry J. Lowe MD¹, Sanjay Malunekar¹, and James Quinn MD, MS²

¹Center for Clinical Informatics. ²Division of Emergency Medicine
Stanford University School of Medicine, Stanford CA

Abstract

Event Stream Processing is a computational approach to the problem of how to infer the occurrence of an event from a data stream in real time without reference to a database. This paper describes how we implemented this technology on the STRIDE platform to address the challenge of real time notification of patients presenting in the Emergency Department (ED) who potentially meet eligibility criteria for a clinical study. The system was evaluated against a standalone legacy alerting system and found to perform adequately. While our initial use of this technology was focused on relatively simple alerts, the system is extensible and has the potential to provide enterprise-level research alerting services supporting more complex scenarios.

Introduction

Event Stream Processing (ESP)ⁱ has in the past decade emerged as an important technology in a variety of applications, from credit fraud detection and financial market analysis to epidemiologyⁱⁱ and clinical alertsⁱⁱⁱ. Computer-based clinical monitoring, event detection and alerting systems have been applied in a number of areas^{iv,v,vi,vii}. This paper describes a real-time clinical event notification system designed to support the needs of clinical and translational research. The system, built on the Stanford Translational Research Integrated Database Environment (STRIDE)^{viii} platform, uses an open source ESP software package (Esper^{ix}) to analyze the clinical HL7 feeds populating the STRIDE Clinical Data Warehouse^x (CDW), to identify patients who may be eligible for clinical research studies. We describe how the system supports an existing clinical study and how we are using it to determine feasibility for future studies.

Methods

Event Stream Processing is a well-established computational approach to the problem of deriving event triggers from data streams. It is designed to permit near instantaneous analysis and response to

the information in a real-time event stream, such as an HL7 clinical data feed. In conventional approaches to the problem of drawing inferences from a series of apparently disjoint events, temporal ordering of events is important, but the computational analysis time required may make real-time alerting difficult to achieve. ESP addresses this issue by eliminating the step of first storing in a database the information to be analyzed, instead placing the inference engine directly in the stream of incoming events. Any additional data needed to support decision-making is typically pre-loaded into memory in the inference engine for performance reasons.

Complex Event Processing (CEP)^{xi,xii} is an offshoot of ESP where the real time data stream is monitored not only for single events such as the admission of a patient to the emergency department with a specific diagnosis but for multiple events in specified temporal relationships, such as the ordering of a head CT for a patient with a traumatic brain injury who may have recently been taking anti-coagulants. Successful notification in these scenarios may require monitoring two or more HL7 message streams. In one of the studies that we supported as part of this system evaluation we monitored the lab order stream (we used an INR order as probable proxy evidence for being on an anti-coagulant) and the radiology order stream (looking for specific types of imaging studies). Notifications were only to be triggered when both orders we were looking for appeared within a given time frame for the same patient.

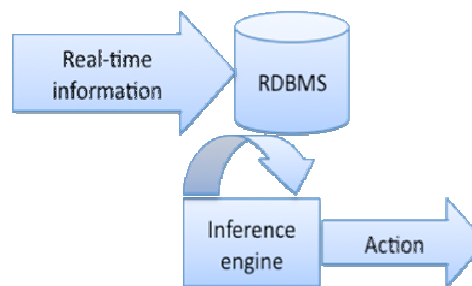


Figure 1: Conventional Database-Driven Approach



Figure 2: Event Stream Processing (ESP) Approach

Software frameworks for ESP and CEP consist of a rules definition language and an inference engine that monitors the incoming event stream looking for patterns specified in the currently active rule set. Sophisticated frameworks also include a user interface for rules definition, and various dashboard views summarizing current and past engine activity.

In selecting Esper over its competitors such as Progress Software, Aleri, StreamBase and Tibco (to name a few), cost was a major determinant. Esper is available under the GPL GNU Public License v2.0 open source license and was the only free product offering with sufficient market recognition to rate a mention in the Q3 2009 Forrester analysis on complex event processing^{xiii}.

The Esper engine is Java based, so integration with a real time data feed takes the form of custom Java classes. The rules engine is driven by a simple variant of SQL. For example, we used the following rule to define the alert trigger for patients who had a head CT scan and an INR blood test (Prothrombin Time International Normalized Ratio) ordered within four hours of each other:

```

select ... from
OrderEvent(code='CT_HEAD' and
visitDeptName='Emergency').win:time(4 hours) as ct,
OrderEvent(code='INR' and
visitDeptName='Emergency').win:time(4 hours) as inr
where ct.visitNumber = inr.accountNumber and ct.mrn = inr.mrn
  
```

The engine is designed to keep in memory only such data as might be needed to trigger currently defined alerts. All of the system's rules are written to use a 4-hour window, so the engine automatically 'discards' (marks as available for garbage collection) information received more than 4 hours ago, thereby preventing memory leaks. Furthermore, we first write each event out to disk prior to feeding it into the engine, and on system start-up the raw data log files are loaded, so in the event of system restart we can re-run all missed messages and generate (admittedly belated) alerts. In the unusual event that our message receiver is also offline, the events are queued in the source system at the hospital and are re-sent when the receiver comes back online. To prevent alert duplication on restart we also maintain a log of successfully delivered notifications.

We evaluated the system using two scenarios for recruiting patients at Stanford University Medical Center (SUMC). The first involved an ongoing NIH study to prospectively evaluate all dog bites in the emergency department in order to develop high-risk criteria for those who may develop infection and thus may benefit from prophylactic antibiotics. The study involves real-time identification and completion of a data form by the treating physicians. Investigators receive notification when a patient arrives and is registered with a complaint of dog or animal bite. The study investigator then calls in and has the treating physician complete a prospective data form in real-time that becomes part of the electronic record and is easily downloaded to the study database. All patients are followed-up at 14 days to determine the presence of an infection. Prospective enrollment of patients in this manner allows accurate collection of all necessary study variables.

The second scenario is a feasibility study to determine the sample size and ability of the system to identify patients with traumatic brain injury on anticoagulation and to test secure messaging. This study required real-time notification of traumatic head injury patients on anticoagulation. This scenario would allow testing of the ability of the system to screen on various HL-7 streams such as lab order (INR), radiology order (CT order) and laboratory test result (elevated INR, or INR > 1.2).

As the investigators involved in the dog bite study had previously implemented a limited standalone system to receive and parse HL7 messages to support this work, and were satisfied with their system's ability to identify patients who might be potential study candidates, we choose to evaluate the new STRIDE-based system using their legacy system as the "gold standard".

As is often the case, we were forced to respond to changes in clinical practice even as we were verifying the system's operation, due to a new clinical system implementation. The old clinical system generated HL7 ADT messages based on data hand-entered during the ER registration process. The strings we looked for in these messages contained the words "animal" or "dog" and "bite", in any combination, in either segment PV2.3 (admit reason) or segment PV2.12 (visit description) of the ADT message. Once the new clinical system was implemented we found that even though the clinical workflow had changed from typing in a string of text to selecting a coded value from a pick-list, the resulting ICD9-coded values were placed in the same message segments and contained much the same text

as before, so we did not need to alter our algorithm in any way during the evaluation.

Our initial implementation was to automatically notify, by pager and email, the study coordinator when patients presented in the ED with a dog or animal bite. No protected health information (PHI) was exposed in this electronic alerting notification process. From a technical perspective, the project involved the overhead of learning the rules language, developing an adapter for HL7 messages, and coordinating a production release of the new software package with the STRIDE operations team. We validated our implementation by running it in parallel with the investigators' legacy notification system for 6 weeks, counting cases where alerts came in from only one source. After this pilot phase was completed successfully the legacy system was decommissioned and the investigators relied on the STRIDE-based system. We then added the capability to deliver secure notifications containing PHI, to further streamline the study enrollment process. Subsequent validation by looking in the clinical system of record for all patients with dog bite as a discharge diagnosis continued to demonstrate 100% accuracy in notification.

We were able to further evaluate the system by rapidly deploying a second research alert. In this study we wished to notify researchers when patients in the Emergency Department were ordered a head CT scan within four hours of an INR order with a result $>1:1$. An alert is generated if a head CT and INR order are found for the same patient within four hours of each other, on the assumption that the ordering of an INR is an indication that the patient is taking anti-coagulants. A second alert is then generated if the results of the INR come back greater than 1.1.

Results

The dog bite study data feed was first piloted in Summer 2009. During a six-week pilot phase we ran both the legacy and STRIDE-based systems in parallel and monitored for any missing alerts. A total of 45 tests alerts were generated during the pilot phase. One alert was not generated by STRIDE. The cause was traced to a faulty email relay, and remediating measures were put in place to ensure the problem would not recur. No false alerts were generated. Since the system went into production in September 2009 seventy-two alerts have been generated, averaging seven per month, as shown in Figure 3.

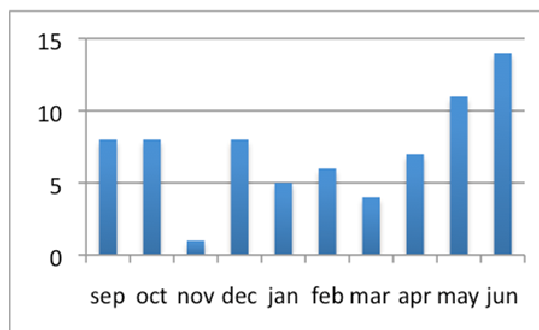


Figure 3: Alert rates for dog bites since the system went into production September 2009, through June 2010.

Since launching the CT head/INR alerts the alert rate has averaged 13 per month, one-third of which trigger a subsequent alert for high INR, as shown in Figure 4.

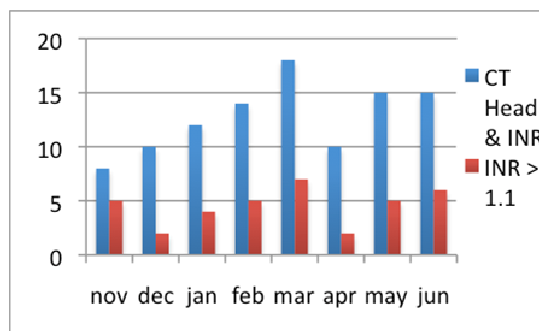


Figure 4: Alert rates for CT head & INR and INR > 1.1 since November 2009, through June 2010.

Analyzing the distribution of the time elapsed between the ordering of the INR and receiving the INR results shows that most results come back in 20 minutes or less, with a few outliers, as shown in Figures 5 and 6. This is an important consideration, given the four-hour time window defined in the alert rule for this study. No INR result - Head CT scan order pair extended beyond the alert's temporal window.

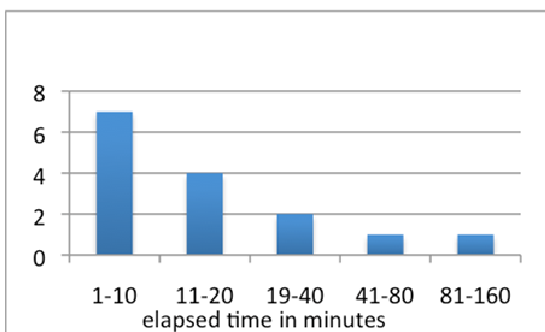


Figure 5: Logarithmic scale graph of elapsed time between ordering the test and receiving the notification of an INR result > 1.1.

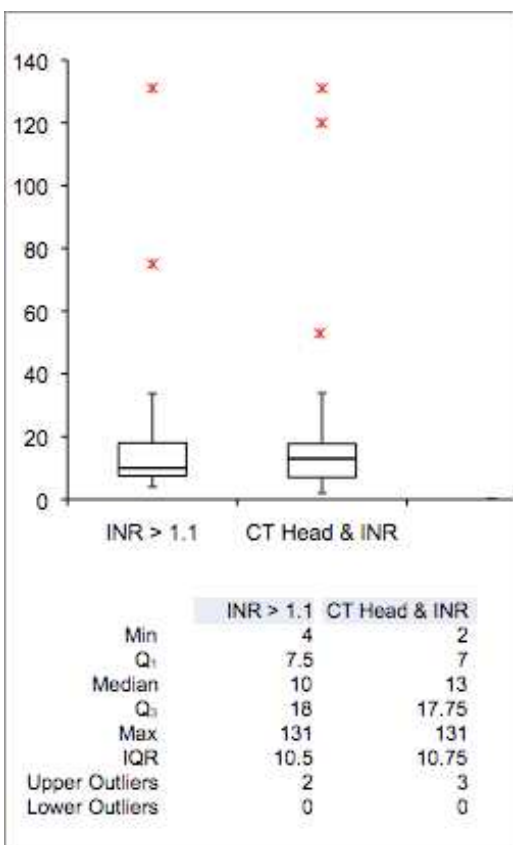


Figure 6: Box plots and accompanying statistics for elapsed time between related CT Head and INR orders

Conclusion

We were able to successfully leverage the Esper open source ESP/CEP engine and an existing HL7 feed using software developers familiar with Java and

with HL7 to very rapidly and cost-effectively deploy a research alerting system on the STRIDE platform. We evaluated the system using two clinical alerts designed to support simple study enrollment and found it to be effective in both use cases.

Given the difficulties that many clinical research studies encounter in recruiting an adequate number of participants^{xiv}, "e-screening" systems^{xv} are of potential importance^{xvi}. Both real-time (generally reflecting a time-dependent need to identify and seek consent from patients for inclusion in a research study) and delayed research alerts (using partial matching to clinical trial eligibility criteria based on data in HL7 messages) can be generated using rule-based ESP/CEP. The system that we describe in this paper currently monitors real-time clinical data feeds using a pure ESP approach, without reference to a database, and we are successfully employing it for automated identification of patients who might be eligible for research studies based on a combination of temporally related clinical criteria. It is of interest to note, however, that the Esper engine lends itself well to database integration. Hybrid systems that use the event stream to watch for post-conditions but then consult a relational database for necessary pre-conditions can be somewhat easier to implement than a pure ESP approach, which can necessitate keeping potentially very large historic data sets entirely in memory. We plan to explore this avenue in the coming year as we have several researcher groups expressing an interest in this enhancement to STRIDE.

Acknowledgements

Supported in part by NIAMS R21 AR054503-02, NINDS U10 NS058929-02 and NCRR UL1 RR025744

References

- i Luckham, David. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*, Addison-Wesley Professional, 2002.
- ii Wagner, M., Espino, J., Tsui, F., Wilson, T., Tsai, T., Harrison, L., and Pasculle, W. "The Role of Clinical Event Monitors in Public Health Surveillance." *Journal of the American Medical Informatics Association*, 2002

iii A. D. Crew, K. D. C. Stoodley, R. Lu, S. Old and M. Ward, "Preliminary clinical trials of a computer-based cardiac arrest alarm", *Intensive Care Medicine*, Volume 17, Number 6 / June, 1991 pp. 359-364

iv McDonald CJ. Use of a computer to detect and respond to clinical events: its effect on clinician behavior. *Ann Intern Med* 1976 Feb;84(2):162-7

v Rocha BH, Christenson JC, Evans RS, Gardner RM. Clinicians' response to computerized detection of infections. *J Am Med Inform Assoc*. 2001;8:117-125.

vi Hripcsak G, Bakken S, Stetson PD, Patel VL. Mining complex clinical data for patient safety research: a framework for event discovery. *J Biomed Inform*. 2003;36:120-130.

vii Chen ES, Wajngurt D, Qureshi K, Hyman S, Hripcsak G. Automated real-time detection and notification of positive infection cases. *AMIA Annu Symp Proc*. 2006:883.

viii Lowe HJ, Ferris TA, Hernandez PM, Weber SC. STRIDE: An Integrated Standards-Based Translational Research Informatics Platform. *AMIA Annu Symp Proc* 2009

ix <http://esper.codehaus.org/>

x <http://clinicalinformatics.stanford.edu/projects/cdw.html>

xi D. Luckham, *The power of event, introduction to complex event processing*, Addison Wesley, 2002.

xii Mendes MR, Bizarro P, Margues P. A framework for performance evaluation of complex event processing systems *Proceedings of the second international conference on Distributed event-based systems*. 2008, Pages 313-316, ACM.

xiii

http://www.forrester.com/rb/Research/wave%26trade%3B_complex_event_processing_cep_platforms%2C_q3/q/id/48084/t/2

xiv Treweek S, Mitchell E, Pitkethly M, et al. Strategies to improve recruitment to randomised controlled trials. *Cochrane Database Syst Rev*. 2010:MR000013.

xv Thadani SR, Weng C, Bigger JT, Ennever JF, Wajngurt D. Electronic Screening Improves Efficiency in Clinical Trial Recruitment. *J Am Med Inform Assoc*. 2009;16:869-873.

xvi Dugas M, Lange M, Müller-Tidow C, Kirchhof P, Prokosch HU. Routine data from hospital information systems can support patient recruitment for clinical studies. *Clin Trials*. 2010;7:183-189.