

APPLICATION OF AN EVENT-TRIGGER-RULE SYSTEM TO AGRICULTURAL HOMELAND SECURITY

Seema Degwekar

Database Research and Development Center
University of Florida, Gainesville, FL – 32611, U.S.A.
spd@cise.ufl.edu

Stanley Y.W. Su

Distinguished Professor, Computer and Information Science and Engineering and Director, Database R&D Center
University of Florida, Gainesville, FL – 32611, U.S.A.
su@cise.ufl.edu

Howard Beck

Professor, Agricultural and Biological Engineering, and R&D Director, Office of Information Technologies, Institute of Food and Agricultural Sciences
University of Florida, Gainesville, FL – 32611, U.S.A.
hwb@ufl.edu

Carrie Harmon

Plant Pathology Coordinator, Southern Plant Diagnostic Network, Department of Plant Pathology, Institute of Food and Agricultural Sciences
University of Florida, Gainesville, FL – 32611
clharmon@ifas.ufl.edu

Gail Wisler

Professor and Chair, Department of Plant Pathology, Institute of Food and Agricultural Sciences
University of Florida, Gainesville, FL – 32611, U.S.A.
gcwisler@ifas.ufl.edu

ABSTRACT

This paper discusses the detection and management of a soybean rust outbreak in the context of agricultural homeland security. An Event-Trigger-Rule system is used for event registration, filtering and notification, and for process coordination and enforcement of agencies' policies, constraints, regulations and data integrity/security/privacy. A 'Response and Action Plan' for combating the disease proposed by one of the 12 member states of the Southern Plant Diagnostic Network is used in a prototype implementation to demonstrate the utility of the system.

KEY WORDS

Event and Rule-based notification, Homeland Security

1. Introduction

Disease detection and control, illegal drug trafficking, border control, illegal immigration, and terrorism are examples of complex problems being faced by all nations. As an aid to solving these problems, government agencies within a country and across national boundaries need to share information in a timely manner, coordinate their

activities, keep one another informed of the occurrences of relevant events, and enforce global and local policies, constraints, regulations, and security and privacy rules.

In this work, we are dealing with the general problem of agricultural homeland security, and specifically, the rapid detection and reporting of crop pest and disease outbreaks that may have been introduced intentionally or unintentionally. We focus on one example, the detection and management of a soybean rust outbreak in the United States, to illustrate our approach. We have developed general mechanisms for event notification, event filtering, process coordination and rule processing for achieving inter-agency knowledge sharing and collaboration. An Event-Trigger-Rule (ETR) System [1] developed at the Database Systems Research and Development Center of the University of Florida is used to define and register events related to soybean rust detection, notify people and agencies of the occurrences of events, and trigger rules to invoke manual and automated processes and enforce agencies' policies, constraints, regulations and data integrity/security/privacy concerns.

The organization of this paper is as follows. Section 2 presents related work in this area. Section 3 introduces the National and State Plant Diagnostic Networks. Section 4 describes our prototype system architecture and Section 5

gives implementation details. Section 6 concludes with some thoughts on future work.

1. Related Work

In this section, we compare our ETR approach with others and note several similarities and differences. In JEDI [2], the authors outline the concepts of event description, subscription, and notification, which were also introduced in ETR. A key feature of GEM [3] and ELVIN[4] is the composition of high-level events from low-level ones, which is similar to ETR's event history (or composite event). PENS [5] is an event-based system that focuses more on predictive algorithms than actual notification. READY [6] uses the CORBA Notification Service. Event actions are limited to notifications. SIP [7] offers the capability of controlling electric devices among other forms of notification. Finally, [8] presents the motivation for event notification. It does not discuss event filtering, which is important for our application.

A major advantage of our system is its management and processing of distributed events and event-triggered rules. Rules originally emerged in the research area of artificial intelligence and expert systems. They were soon incorporated into databases to form active databases [9], (e.g. ARIEL [10], SENTINEL [11]). The use of rules in a distributed environment has been researched to achieve site autonomy as well as rule performance increase [12]. The addition of rules to our event-based system brings in these advantages. Overall, ETR includes most of the features present in these systems. In addition, it also has key assets like distributed event and rule definitions and management, complex event relationships and rule processing, dynamic rules, and specification of synchronous and asynchronous events.

The above discussion is pertinent to event-based systems in general. We now focus on work related to the domain of agricultural security. Most research, extension services and farmers use email alerts, newsletters or text messages as the primary forms of receiving/disseminating information ([13], [14]). To the best of our knowledge, there are currently no event-based/rule-based notification systems in the agricultural domain. The techniques used in the systems mentioned above require interested parties to seek the information they need by logging-in on a website. Also, state/national agencies generally send out alerts only in cases of a severe situation. With ETR, farmers, extension agents, etc. would be able to specify the information they are interested in and define rules that will be automatically fired to perform a variety of actions when the required information is available. As we shall show, specification of rules is much more powerful than a simple notification system.

Currently, there are many expert and automated systems ([15], [16], [17], [18]) that aid in diagnosis for different kinds of pests. However, they have not been integrated with mechanisms for event definition, registration, filtering and notification.

2. The National and State Plant Diagnostic Networks

The United States Department of Agriculture, Cooperative State Research, Education and Extension Service (USDA, CSREES) launched a multi-year national project in May 2002 to build the National Plant Diagnostic Network (NPDN) [19]. This was done with a view to strengthen the homeland security protection for the nation's food and agriculture by facilitating quick and accurate detection of disease and pest outbreaks in crops. Such outbreaks can occur naturally as foreign pathogens are introduced into the United States through mechanisms ranging from accidental importation to air-borne introduction by currents that traverse entire continents. They can also occur intentionally through deliberate introduction, as an act of bioterrorism [20]. In this nationwide effort, the University of Florida was selected as the southern regional center of NPDN and is responsible for building a Southern Plant Diagnostic Network (SPDN). SPDN consists of a regional center system at the University of Florida and 12 state systems (counting Florida) and Puerto Rico connected to the regional center system through the Internet. The functions of the SPDN Regional Center are to establish a network for the detection and diagnosis of plant health problems, extend and support sound public policies, implement environmentally sound prevention and management strategies, and provide leadership and training. Data collected from the region is sent to the National Agriculture Pest Information System at the Purdue University in Indiana.

In May 2004, the SPDN states participated in a workshop centered on the detection and management of soybean rust. Asian soybean rust [21] is an aggressive fungal disease that can potentially reduce soybean yield by as much as 80 percent. After it was first recorded in Japan in 1902, the pathogen moved through Asia, Australia, and Africa before it was discovered in South America in 2000. It has now been detected moving northward through South America. The threat of the rust entering the United States is very real and hence SPDN is investigating mechanisms for its detection and management. While the federal government has not imposed an eradication policy against soybean rust, US will need to decide on a state-by-state basis which management measures to employ to protect its agricultural economics. It is believed that a communication and notification network and a response plan in each state are required.

This paper represents a prototype system to demonstrate the application of the ETR technology to the detection and management of soybean rust outbreaks. At the Database Systems Research and Development Center of the University of Florida, we have developed an Event Manager and an Event-Trigger-Rule (ETR) Server. The former is used for defining and managing distributed events and event filters, and performing event notifications associated with a potential soybean rust

<p>1) Event : Suspect_Sample_In_System Provider : NPDN Triage Lab Subscribers : Origin State APHIS SPHDⁱ ,Origin SPROⁱⁱ Rules :</p> <p>a) Prepare a response by collaboration between SPHD and SPRO.</p> <p>b) For Origin SPRO, also notify surrounding SPROs and tell them to be prepared too.</p> <p>2) Event : Preliminary_Diagnosis Provider : USDA Expert Lab Subscribers : Origin NPDN Regional Director, Expert Lab's SPROⁱⁱⁱ ,Expert's SPHD^{iv} Rules :</p> <p>a) For NPDN Regional Director, forward the event to NPDN Program Manager and other NPDN Regional Directors. Also forward it to regional diagnostic labs without disclosing the state of origin of the sample.</p> <p>b) For SPRO and SPHD, be aware of sample in system. Raise alert level.</p>	<p>3) Event : Confirmed_Diagnosis Provider : APHIS Confirming Diagnosis Designate^v Subscribers : Origin State APHIS SPHD, NPDN Triage Lab Rules :</p> <p>a) For SPHD, forward to origin state SPRO. Initiate response.</p> <p>b) For Triage Lab, record this in the database. Forward results to USDA Expert Lab^{vi}.</p> <p>4) Event : Soybean_Rust_Confirmation Provider : APHIS Confirming Diagnosis Designate Subscribers : USDA Expert Lab, Origin State APHIS SPHD, NPDN Triage Lab Rules :</p> <p>a) Notify all key state and external stakeholders (i.e. SPROs and SPHDs) and send out public statement within an hour of event generation.</p> <p>b) Release soybean rust fact sheets to press on day 1.</p> <p>c) Activate the black section on www.soybeanrust.info to provide more details on the disease outbreak and the level of preparedness against it.</p>
<p>i. Animal and Plant Health Inspection Service's State Plant Health Diagnostic lab of the state of origin of sample</p> <p>ii. State Plant Regulatory Official of the state of origin of sample.</p> <p>iii. State Plant Regulatory Official of the state of the Expert Lab</p> <p>iv. State Plant Health Diagnostic lab of the state of the Expert Lab</p> <p>v. Animal and Plant Health Inspection Service's designate - the final authority on confirming any diagnosis.</p> <p>vi. United States Department of Agriculture's Expert Lab</p>	

Figure 2: Examples of triggers

template address to the APHIS Lab first, waits for the lab official to fill out their part, and then notifies the SPRO to complete the response. Another interesting event is “Soybean_Rust_Confirmation”, which corresponds to the identification of a diseased sample. One of the rules involved is to notify the key and external stakeholders and send out a public statement within the hour of posting the event. Another involves sending out fact sheets and activating a website section on the first day.

4. Implementation Details

Our rule model [1] is an extension of the traditional object model. It consists of three components: events, rules, and triggers. An *event* is an occurrence of any incident of importance. Events can be grossly categorized into three types: a) method-associated events that are raised before and/or after the execution of a specific method of an object class, b) explicitly posted events (i.e., events posted by programs when some internal data states have been reached), and c) timer events (events posted by a timer at a certain time or at/within a specified time interval). The next component of the model is a *rule*. A rule is a high-level specification of a granule of executable code that is related to an event or events. It is composed of condition, action and alternate action clauses. Each rule can have a guarded condition expression, which consists of a number of guards and a guarded expression. If, during the condition evaluation, any guard is evaluated to false, the

entire rule is skipped. (This feature is useful as under some conditions the rule may not be applicable.) If all the guards are evaluated to true, and the final expression is evaluated to true also, the guarded condition expression is true and the action clause is processed. If all the guards are evaluated to true, and the guarded expression is evaluated to false, the guarded condition expression is false and the alternative action clause is processed. The third component is the *trigger*, which basically specifies which distributed events can trigger which distributed rules. The syntaxes of the definition languages for events, rules, and triggers can be found in [1]. In this paper, we will illustrate the syntax with examples given in the previous section.

The ETR system is composed of three sub-systems – Event Manager, ETR Server, and Profile Manager. The Event Manager deals with accepting event registrations and delivering the events to subscribers of the event. The ETR Server executes triggers and rules at run time. It can schedule complex rule structures specified in a trigger. The Profile Manager provides a graphical user interface for event providers and subscribers to define/browse events, rules, and triggers. Events and rules can be defined by different agencies and stored in a distributed fashion and managed and processed by replicas of the Event Manager and the ETR Server installed at various sites. Access and use of each of these is protected for security reasons. Like events and rules, triggers are stored at the sites at which they are defined. Another feature of

EVENT	SBR*_Confirmation(String sampleID, String origin, String stage_of_advancement)
DESC	This event is generated when a suspect soybean sample turns out to be affected with the soybean rust disease. If this event occurs, key stakeholders are notified and press releases are sent.
TYPE	EXPLICIT
RETURNS	String

Figure 3: Event “Soybean_Rust_Confirmation”

TRIGGER	SBR*_Confirmation_Trigger (String sampleID, String origin, String diseasestage)
TRIGGEREVENT	SBR*_Confirmation
RULESTRUC	SBR*_Confirmation_Rule
RETURNS	String : SBR*_Confirmation_Rule

Figure 5: “Soybean_Rust_Confirmation_Trigger”

* SBR corresponds to Soybean Rust in Figures 3 – 5.

the ETR system is Event History Processing. A trigger specifies an event structure, which is composed of two parts: TRIGGEREVENT and EVENTHISTORY. The TRIGGEREVENT specifies a number of alternative events, each of which, when posted, would trigger the evaluation of the event history. EVENTHISTORY is a complex expression that states the relationship between the events that have been posted. When an event specified in TRIGGEREVENT occurs, the event history is evaluated. If it evaluates to true, the rule structure is fired. This can be a linear, tree or network structure of rules given in the RULESTRUC clause.

Consider “Soybean_Rust_Confirmation”. The event definition is given in Figure 3. The ‘type’ of the event denotes whether it is generated by some method (programmatically), posted explicitly (as in our case), or posted by a timer. The other clauses are self-explanatory. The corresponding rule for the event is shown in Figure 4. A dynamic rule can be modified during run-time, whereas a static rule cannot. The RULEVAR clause defines the rule variables used in the ACTION clause. Although our rule implementation uses Java as its underlying programming language, we have shown these rules in simple English here to make it more readable. The EXCEPTION clause specifies any possible exceptions that can arise during rule execution.

The times of 60m (60 minutes), and 24h (24 hours) associated with some actions denote the time constraints during which these actions must be taken. The corresponding trigger for the event-rule pair discussed above is given in Figure 5. The event and the rule are parameterized. The trigger maps event parameters to rule parameters and returns results generated by the rule to the program that posted the event.

Figure 6 illustrates the entire process of event-trigger-rule definition and processing. First, using Web-based graphical user interfaces (GUIs), the event “Soybean_Rust_Confirmation” and the corresponding

RULE	SBR*_Confirmation_Rule(String sampleID, String origin, String diseasestage)
RETURNS	String
DESC	This rule is activated when the event “Soybean_Rust_Confirmation” is raised. Certain actions need to be taken within a pre-specified time constraint.
TYPE	DYNAMIC
STATE	ACTIVE
RULEVAR	String SubscriberList, String statement
ACTION	1) Notify(SubscriberList, 60m) and Send_Public_Statement(60m) 2) Send_Fact_Sheet(SPRO, 24h) 3) Activate (www.soybeanrust.info/todo, 24h)
EXCEPTION	Notification Undelivered. Handler: Notify admin. Resend.

Figure 4: Rule “Soybean_Rust_Confirmation_Rule”

rule “Soybean_Rust_Confirmation_Rule” are defined. The event provider can then register his/her event with the ETR system to make it accessible to others. An interested party (the SPRO in Florida, for example) can then subscribe to this event. During this process, the user is asked to give at least a verified email address for contact. This is to ensure that no untraceable subscribers exist.

In addition, other contact information such as a cell phone number (for text message notification), or a remote program (to carry out initial automated tasks, for example) may also be given. At this time, the subscriber is also asked to provide values for the event parameters specified in an event filter template. The filled template forms the event filter of the subscriber. If the specified values match with the data associated with the event when it occurs (i.e., the event data), the subscriber will be notified and the event data will be delivered to him/her. If any field of the event filter template is left blank, it indicates that the subscriber does not have any preference for the field values (i.e. that field does not participate in event filtering). This completes the registration-subscription process.

Depending on the event type, it is either automatically generated or posted by the provider. We have chosen the “Soybean_Rust_Confirmation” event to be explicitly posted by the provider. During the event posting, the provider is asked to fill in the event data, which are used for matching with the subscriber’s preferences. Only those subscribers whose preferences match the event data will be notified. This cuts down unnecessary notifications and thus resource consumption. Notifications are sent in all forms of communication listed by the subscriber. In the figure, the subscriber wishes to be notified by email, text message on a cell phone, and invocation of a program on his/her machine. The rule associated with the event carries out all the actions mentioned in Figure 5. We have implemented the system using Java and Java servlets with Apache Tomcat v4.1.3

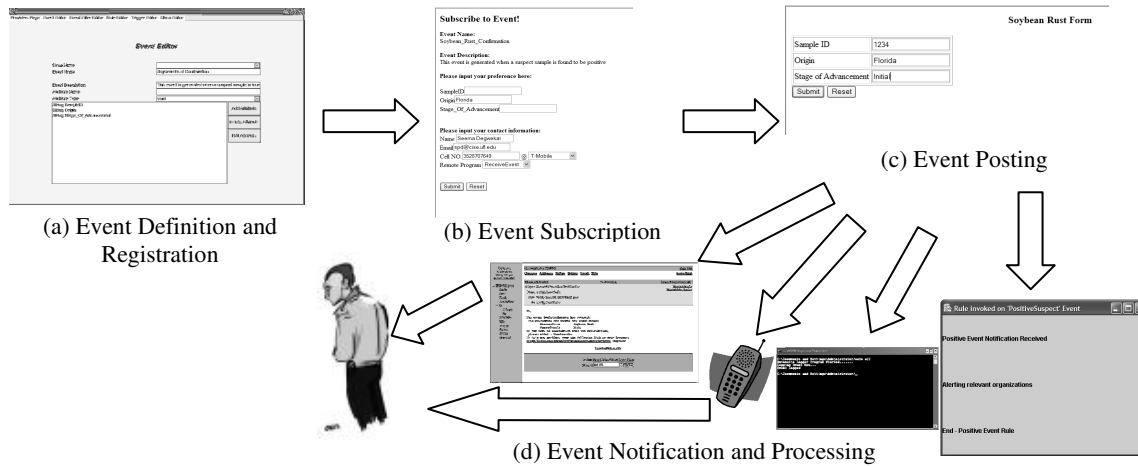


Figure 6: Event Definition, Registration, Subscription and Handling

as the web server. We use the technologies of web services and RMI for subscription and notification. We also use the “role” feature in Tomcat to restrict access to certain system features.

5. Conclusion and Future Work

In this paper, we have demonstrated the need of an event-trigger-rule-based system like the ETR system in the field of agricultural disease detection and management. The initial prototype looks promising and we are making some enhancements. First, we are looking at improving the user interface. Currently, the ACTION and ALTERNATIVE_ACTION clauses of rules are written in Java. We are developing a high-level editor for rules that will free rule providers from having to use a particular programming language’s syntax and semantics. Also, current notification options are limited to email and cellular phones. We would like to look into the possibility of including other devices such as pagers, printers, sensors, alarms, etc. Thus, along with the implemented event and rule mechanisms, these devices can also carry out predefined tasks that will respond more efficiently and effectively to various situations.

4. Acknowledgements

Research reported in this paper is funded by USDA Award No. 2002-30001-12082. Any opinion, findings, and conclusion expressed in this paper are those of the authors and do not necessarily reflect the views of USDA.

References:

[1] M. Lee, S.Y.W. Su, & H. Lam, Event and Rule Services for Achieving a Web-based Knowledge Network. Technical Report, Computer & Information Sciences & Engineering, University of Florida, 2000.
 [2] G. Cugola, E. Di Nitto, & A. Fuggetta, The JEDI event-based infrastructure and its application to the development of the OPSS WFMS, *IEEE*

Transactions on Software Engineering, 27(9), 2001, 827 – 850.
 [3] M. Mansouri-Samani, & M. Sloman, GEM: A generalized event monitoring language for distributed systems, *IEEE/IOP/BCS Distributed Systems Engineering Journal*, 4(2), 1997, 96 – 108.
 [4] R. Robinson & A. Rakotonirainy, Multimedia customization using an event notification protocol. *Proc. 22nd International Conference on Distributed Computing Systems Workshops*, Vienna, Austria, 2002, 549 – 554.
 [5] Jeng Jun-Jang & Y. Drissi, PENS: a Predictive Event Notification System for e-commerce environment. *Proc. 24th Annual International Computer Software and Applications Conference*, Taipei, Taiwan, 2000, 93 – 98.
 [6] R. Gruber, B. Krishnamurthy & E. Panagos, READY: a high performance event notification service. *Proc. 16th International Conference on Data Engineering*, San Diego, CA, 2000, 668 – 669.
 [7] K. Arabshian & H. Schulzrinne, A SIP-based medical event monitoring system. *Proc. 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry*, Santa Monica, CA, 2003, 66 – 70.
 [8] V. S. W. Eide, F. Eliassen, O. Lysne, & O. Granmo, Real-time processing of media streams: a case for event-based interaction. *Proc. 22nd International Conference on Distributed Computing Systems Workshops*, Vienna, Austria, 2002, 555 – 562.
 [9] J. Widom & S. Ceri, *Active Database Systems: Triggers and Rules for Advanced Database Processing* (Morgan Kaufmann, CA, 1996).
 [10] E. Hanson, The Design and Implementation of the Ariel Active Database Rule System, *IEEE Transactions on Knowledge and Data Engineering*, 8(1), 1996, 157 – 172.
 [11] S. Chakravarthy, V. Krishnaprasad, E. Anwar, & S. Kim, Composite Events for Active Databases: Semantics, Contexts and Detection. *Proc. 20th*

International Conference on Very Large Databases, Santiago de Chile, Chile, 1994, 606 – 617.

- [12] S. Ceri & J. Widom, Production Rules in Parallel and Distributed Environments. *Proc. 18th International Conference on Very Large Databases*, Vancouver, Canada, 1992, 339 – 351.
- [13] P. Suamsiri, A. Jintrawet, H. Umekawa, R. Ito, & T. Kameoka, A Web-based Information Delivery System for Appropriate Technology for Reduction of Agrochemical in Northern Thailand. *Proc. 4th*
- [14] *International Conference of The Asian Federation of Information Technology in Agriculture and the 2nd World Congress of Computers in Agriculture and Natural Resources*, Bangkok, Thailand, 2004, 154 – 159.
- [15] S. Wilkinson, S. Sim, Y. Suansook, eNACA : A Digital Knowledge-Sharing Strategy for Aquaculture. *Proc. 4th International Conference of The Asian Federation of Information Technology in Agriculture and the 2nd World Congress of Computers in Agriculture and Natural Resources*, Bangkok, Thailand, 2004, 670 – 675.
- [16] B. Mahaman, P. Harizanis, I. Filis, E. Antonopoulou, C. Yialouris, & A. Sideridis, A diagnostic expert system for honeybee pests, *Computers and Electronics in Agriculture*, 36(1), 2002, 17-31.
- [17] C. Yialouris, H. Passam, A. Sideridis, & C. Metin, VEGES – A multilingual expert system for the diagnosis of pests, diseases and nutritional disorders of six greenhouse vegetables, *Computers and Electronics in Agriculture*, 19(1), 1997, 55-67.
- [18] M. Kramers, C. Conjin, & C. Bastiaansen, EXSYS, an Expert System for Diagnosing Flowerbulb Diseases, Pests and Non-parasitic Disorders, *Agricultural Systems*, 58(1), 1998, 57 – 85.
- [19] Y. Koumpouros, B. Mahaman, M. Maliappis, H. Passam, A. Sideridis, & V. Zorkadis, Image processing for distance diagnosis in pest management, *Computers and Electronics in Agriculture*, 44(2), 2004, 121 – 131.
- [20] M. Zhong, Individual Study Report, Computer & Information Sciences & Engineering, University of Florida, 2003.
- [21] J. Tucker, Historical Trends Related to Bioterrorism: An Empirical Analysis, Special Issue, *Emerging Infectious Diseases*, Centers for Disease Control and Prevention, 5(4), 1999, 498 – 504.
- [22] Iowa State University, Iowa Soybean Association and Iowa Soybean Promotion Board, Iowa Department of Agriculture and Land Stewardship, & United States Department of Agriculture – Animal and Plant Health Inspection Service – Plant Protection and Quarantine, Iowa Response and Action Plan for Asian Soybean Rust, Draft 1.5, 2004.