A Distributed Event-triggered Knowledge Sharing System for Agricultural Homeland Security

Seema Degwekar[1], Jeff DePree[1], Howard Beck[2], Stanley Y. W. Su[1]
[1]Database Systems R&D Center
458 CSE Building, University of Florida, Gainesville, Florida 32611
1-352-392-2693
{spd, jdepree,su}@cise.ufl.edu
[2]Department of Agricultural and Biological Engineering,
University of Florida, Gainesville, Florida 32611
1-352-392-3797
hwb@ufl.edu

Abstract—*Government agencies globally are facing problems like illegal immigration, terrorism, and disease diagnostics and control. Solutions to these problems rely heavily on collaborating organizations' ability to effectively and efficiently share not only* ***data*** *but also* ***knowledge*** *embedded in organizational and inter-organizational policies, regulations, data and security constraints, processes and procedures. The United States Department of Agriculture has launched a multi-year national project to build the National Plant Diagnostic Network (NPDN) for strengthening the homeland security protection of food and agriculture by connecting five regional plant diagnostic centers with a national center. Complementing this effort, our research team has been developing a web-based, distributed system for event-triggered knowledge sharing among NPDN organizations. We capture multi-faceted knowledge using three types of rules and rule structures. A user-friendly interface is provided for collaborating organizations to define events of interest as well as knowledge rules and publish them in a global registry for browsing, querying, event subscription and notification, and processing. Event data are the* ***dots*** *that can be* ***connected across organizational boundaries*** *through the interoperation of knowledge rules and rule structures.*

1. INTRODUCTION

Effective resource sharing, collaboration and coordination among government organizations is needed to solve complex problems related to homeland security such as border control, illegal immigration, terrorism and bio-security threats. Each complex problem has to be tackled from social, economical, political, moral, as well as technical perspectives. An individual government organization may not possess the information and know-how required to solve a complex problem alone, but if collaborating organizations can pool their resources together, they can certainly take a big step towards solving it. They need to share not only data, but also human and organizational **knowledge** useful for decision support, problem solving and activity coordination. The technology of sharing distributed, heterogeneous data has been extensively studied, but an effective way of sharing knowledge among collaborating organizations is still lacking.

In this work, we are interested in capturing, managing and applying the **multifaceted knowledge** embedded in organizational as well as inter-organizational policies, regulations, constraints, processes and operating procedures. A common means of representing knowledge is to use knowledge rules [10]. Three types of knowledge rules have been found to be useful in many applications [9, 4]: integrity constraints [11], logic-based derivation rules [12], and action-oriented rules [14]. If they are incorporated in a single knowledge specification language and used together in an interoperable fashion, they can provide a very powerful facility for, not only knowledge representation, but also knowledge sharing. By rule interoperability, we mean that rules of different types can interface with each other seamlessly. For example, a logic-based derivation rule may deduce some data for use by an action-oriented rule, which may activate an application operation to update some data that need to be verified by a constraint rule. In our work, we use these three types of rules to specify organizational and inter-organizational policies, regulations and data/security constraints, and structures of these rules to specify organizational and inter-organizational processes and operating procedures.

Sharing useful data is also an important aspect of collaboration. Since each government agency takes the utmost care about the security of its database, opening it up in its entirety for access by other organizations is not an option. Apart from the security reasons, it is typically not necessary to allow full access to a database in most practical cases. When organizations collaborate, it is usually to solve specific problems. Thus it is important to devise a framework that allows them to share only those data and knowledge pertaining to those problems. Our approach for achieving data and knowledge sharing is to augment an event subscription and notification system with knowledge sharing and processing capabilities. An event is anything of significance to collaborating organizations (e.g. an arrest, a terrorist incident, the detection of a disease, a special state of a database, a signal from a sensor, etc.) that occurs at a particular point in time. Collaborating organizations would obtain only the data that are pertinent to the occurrence of an event of common interest (i.e., **event data**) and process only those knowledge rules that are applicable to the event data. Initially, the event data set contains only the data associated directly with the event occurrence. However, it may change as relevant rules and rule structures are applied on the data set; new data can be generated and old data can be modified, thus making some other knowledge rules and rule structures applicable. Multiple rounds of event data transmission, aggregation and rule processing may take place in order to produce all the data pertinent to the event occurrence. Thus, an event-triggered data and knowledge sharing system that facilitates event subscription, event notification, delivery of event data, and processing and interoperation of applicable knowledge rules and rule structures would be ideal for any collaborative federation and is therefore the focus of our R&D work.

One of the important issues to be addressed is the interoperability of heterogeneous rules. One possible approach is to choose one rule type as a common format and convert the other rule types into this chosen representation [2]. This approach sounds attractive because it only needs a single rule engine to process all the converted rules. However, since different types of rules have significant semantic disparities, converting a rule from one representation to another may lead to loss of information. Another possible approach is to build wrappers around different types of rule engines [8], and provide a common interface to enable these rule engines to exchange data generated by rules. In our opinion, this approach is not ideal either because it will result in a very complex rule system that is difficult to operate and maintain. Instead, we take the approach of translating each rule to program code and wrapping it as a web service for uniform definition, browsing, and processing. Our observation is that rules of different types interface with each other through the data they produce/consume. It is therefore important to provide a uniform mechanism of specifying input and output data for a rule. By converting a rule to a *program function*, we achieve this at the programming level without having to use multiple rule engines or to perform rule-to-rule conversions.

The intended contributions of this paper are:

a) Introducing an effective mechanism of knowledge representation and sharing of distributed heterogeneous rules and rule structures, and
b) Presenting the architecture of an event-triggered knowledge sharing system, and the distributed event and rule processing strategy by using agricultural homeland security as the application domain.

Due to lack of space, we are unable to include a description /comparison of the existing systems and approaches related to our work. Interested readers may refer to [5] for the same.

2. NATIONAL PLANT DIAGNOSTIC NETWORK

The collaborative federation that serves as our application domain is the National Plant Diagnostic Network (NPDN [6]). The U.S. Department of Agriculture (USDA) launched a multi-year national project in May 2002 to build NPDN to link plant diagnostic facilities across the United States. This was done to strengthen the homeland security protection of the nation's agriculture by facilitating quick and accurate detection of disease and pest outbreaks in crops. Such outbreaks can occur as foreign pathogens are introduced into the U.S. either through accidental importation, by wind currents that traverse continents, or by an intentional act of bioterrorism [13]. NPDN aims to achieve its mission by creating a functional nationwide network of public agricultural institutions with a cohesive, distributed system. The system will have the ability to quickly detect deliberately introduced, high-consequence, biological pests and pathogens into agricultural and natural ecosystems of the United States. It provides the means for quick identification and establishes protocols for immediate reporting to appropriate responders and decision makers.

The network allows land grant universities' diagnosticians and faculty, state regulatory personnel, and first detectors to communicate information, images, and methods of detection efficiently and in a timely manner. The network has a hierarchical structure, with NPDN at the top. Directly reporting to NPDN, are the regional level organizations, which form the second level of the hierarchy. Lead universities have been selected and designated as Regional Centers to represent 5 regions across the country, and they are located at Cornell University (Northeast region), Michigan State University (North Central region), Kansas State University (Great Plains region), University of Florida (Southern region), and University of California at Davis (Western region). The third and last level of this hierarchy consists of individual diagnostic labs within the states that report to the assigned Regional Center.

Information about plant samples collected by or submitted to any of the state labs are analyzed, and the lab diagnoses are sent to NPDN national data center at Purdue University. The National Agricultural Pest Information System (NAPIS) of Purdue University has been designated as the central repository for archiving the collected data from the regions. Sample collection is done routinely. Member labs diagnose pests/diseases observed on the sample and report to NPDN. Occasionally, all labs will be put on alert and charged with looking for particular pests/diseases, called "select agents". Ideally, each region will have a standard operating procedure (SOP) to combat a pest-of-concern situation, which details the steps to be taken when such a bio-security event takes place.

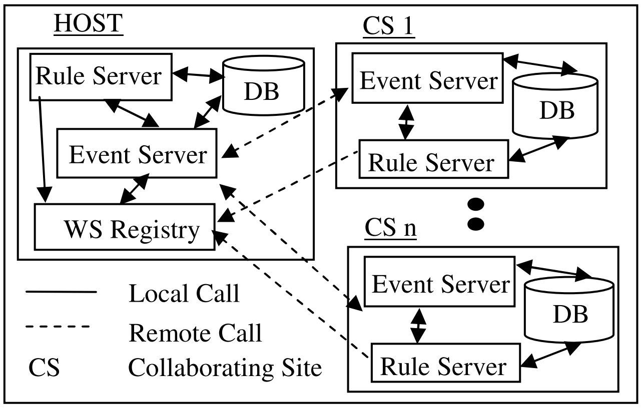

Figure 1. System Architecture

As these diseases can spread rapidly from region to region, there is an overwhelming need for different regions as well as organizations within those regions to communicate and coordinate when a select agent is found. Plant samples and diagnoses need to be transmitted to many organizations for analysis in order to detect any abnormal trend early on. At present, NPDN organizations are developing operational procedures that are, in most cases, carried out manually. There is a need for automated event notification, event data delivery, distributed processing of multifaceted knowledge, and activation of application systems' operations.

3. ARCHITECTURE

The event-triggered knowledge sharing system has a peer-to-peer server architecture, as shown in Figure 1. All participating organizations have identical subsystems installed at their sites (see the main component servers in Figure 1). These include the Event and Rule Servers, as well as a local repository. In addition to these common components, we also need a repository to store information about the registered events, rules, rule structures and operations. This central repository is located at a "host" site, as shown in the figure.

Each collaborating site creates and manages its own events, rules, rule structures and triggers. A *rule structure* can be defined as a directed graph with different types of rules as nodes. These nodes are connected by **link, split**, **and-join**, and **or-join** constructs. A **link** relationship between two rules *r* and *s*, states that rule *r* must be executed before rule *s*, if the link is from *r* to *s.* A **split** relationship between a rule *r* and a collection of rules $s_1, s_2, \ldots s_n$, states that after executing rule *r,* rules $s_1, s_2, \ldots s_n$ can begin their execution in parallel. An **and-join** relationship between a collection of rules $r_1, r_2, \ldots, r_n$ and rule *s* states that rule *s* may begin its execution only after $r_1, r_2, \ldots, r_n$ complete theirs. An **or-join** relationship between a collection of rules $r_1, r_2, \ldots, r_n$ and rule *s* states that rule *s* may begin its execution after a specified number of rules in the collection complete theirs. The number of rules that *s* needs to wait for is specified in the construct. A **trigger specification** connects a distributed event to the distributed rule or rule structure that would be triggered by the event.

When an organization defines a **shared event**, the Event Server at that site stores the event information in the local database and registers this event with the host site. A **local event** is only stored at that site and not registered with the host site. The Event Server manages information about events defined at that particular site and information about event subscribers. If an event occurs at a particular site, the Event Server at that site serves as the coordinator for that particular knowledge sharing session. It carries out event notification by sending event data to sites that have applicable rules and rule structures. The Event Server also handles the aggregation of new event data returned from collaborating sites.

When a shared rule or a rule structure is defined by an organization, the Rule Server at that site converts it into program code, wraps the code as a web service, stores the rule information in the local database and registers the generated web service with the Registry at the host site. A local rule or rule structure is not registered with the host site. The Rule Server is responsible for processing all applicable rules and rule structures when an event occurs.

4. DISTRIBUTED EVENT AND RULE PROCESSING

Each region in the NPDN system is working towards finalizing a set of standard operating procedures (SOPs), each of which will detail the exact procedures to be followed when a select agent is observed. To ensure that appropriate protocol is observed, NPDN has made available a general SOP document [7] to be used by each region as a template for defining its own SOP. In this section, we pick a subset of the communication path outlined in the general SOP and use it as an example to explain our event and rule processing strategy.

Figure 2 depicts the following scenario. When a suspect sample is submitted to NPDN Triage Lab, an event **Presumptive Positive Sample Observed** is said to have occurred. In the figure, we label this event as **E1** for brevity. The occurrence of E1 is denoted as step 1 in the figure. E1 at NPDN Triage Lab causes the event data containing the sample information to be sent to APHIS Lab and NPDN Regional Hub Lab (step 2). This event data is sent to both sites as an XML document. XML is used as a uniform means of information interchange. These labs have applicable rules that can make use of the event data to provide some more relevant information. APHIS (Animal Plant Health Inspection Service) [1] Lab performs confirming diagnosis on the sample (step 3). NPDN Regional Hub Lab informs the appropriate personnel of the sample status (also step 3). These procedures at APHIS Lab and NPDN Regional Hub Lab can be modeled using heterogeneous rules and rule structures. Details of these procedures are given later in this section.

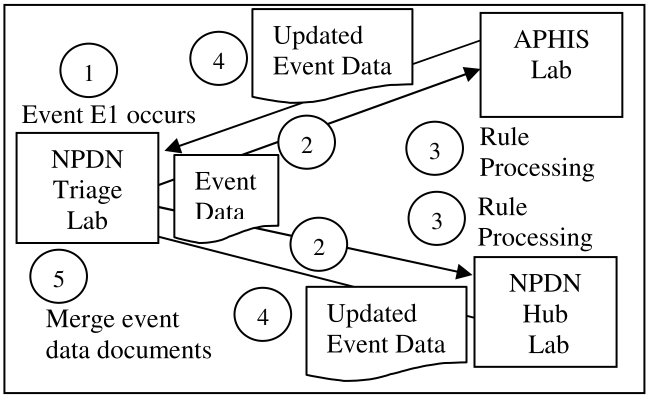

Figure 2. Distributed Event and Rule Processing

The invocation of the applicable rules and rule structures in both APHIS Lab and NPDN Regional Hub Lab may produce new data or modify the existing data. The new data and updates are then sent back to NPDN Triage Lab as modifications to the original event data document (step 4) and are merged with the original event data (step 5). The new version of event data is sent (not shown in the figure) to APHIS Lab and NPDN Regional Hub Lab to begin a second round of rule processing if they have applicable rules.

After all sites have applied their applicable rules and rule structures, the final version of the event data document would contain all the data pertaining to the event occurrence. All applicable sites would receive the final document, which can be used for further decision-making and problem solving.

Figure 3 describes the rule structure executed at APHIS Lab. This is a sequential structure of four rules. APHIS Lab is responsible for conducting a confirming diagnosis on the submitted sample. The first rule, AR1, is concerned with acknowledging the receipt of a presumptive positive sample to NPDN Triage Lab. It also determines the expected date and time of the notification of results. Rule AR2 checks if confirmation will be more than 7 days later and, if so, it consults the APHIS Program Manager to determine whether or not the Regional NPDN Director of the affected region should be contacted. If it is determined that the Director should be contacted, rule AR3 notifies the NPDN Regional Director of the presumptive positive sample in the system as well as other NPDN regional directors. Rule AR4 is concerned with performing the confirming diagnosis, and returns the results to the rule server at APHIS. This rule server then returns the results to NPDN Triage Lab.

Figure 3 also describes the rule structure executed at NPDN Regional Hub Lab. This lab is responsible for employing a local expert to perform the diagnosis and reporting results back to Triage Lab. The first rule, NR1, acknowledges the receipt of the presumptive positive sample by NPDN Triage Lab. Rule NR2 is concerned with asking a local expert to perform some preliminary diagnosis on the sample. Rule NR3, which is processed in parallel with NR2, contacts other personnel such as a State Plant Regulatory Official, and a State Plant Health Director to inform them of the

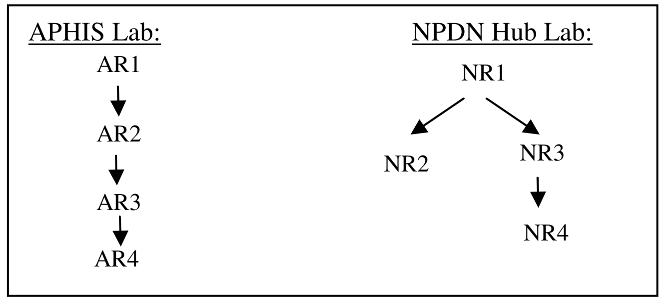

Figure 3. Rule Structures (APHIS, NPDN Hub labs)

presumptive positive sample in the system. It is also concerned with determining whether or not the Regional Hub Lab needs to send the sample to APHIS Lab for further confirmations. If APHIS Lab requires the Regional Hub Lab to send the sample, rule NR4 is concerned with ensuring that the sample is sent and the proper shipping procedure is followed.

Some operations in the rules described above are manual operations. They need to be done by a lab staff, and cannot be automated. One example is the process of diagnosing a sample. Our approach of incorporating manual operations is to require all such operations to be registered with the system prior to them being used in a rule. During the registration process, it is necessary to specify the **agent** who will carry out a manual operation along with his/her contact information that may include either email or cell phone or both. When a rule with a manual operation begins execution, the agent is contacted using either email or text message sent to his/her cell phone, or both, with a message that tells him/her what operation to perform. The instruction also includes how to let the event-and-rule system know when the operation has been performed. Unlike an automated operation, we cannot ensure that a manual operation will be performed as soon as the instruction for performing it is sent. Also, the data generated by a manual operation (e.g., the diagnosis result) may not be available until some time after the request to perform the operation was sent. Through our user interface, the agent can report to the system that the operation has been done and provide its results. Personal notification is not only a step towards ensuring that the operation gets done, but also a way of checking that proper procedures were followed and identifying who was responsible.

5. IMPLEMENTATION DETAILS

We use Java, Sun Java System Application Server 9.0, Enterprise JavaBeans 3.0, the Apache jUDDI project, MySQL 5.0, and AJAX technologies to implement our prototype system. A user interface has been developed to allow collaborating organizations to define their events, rules, rule structures, triggers, and manual and automated operations used in rules. Display pages of the user interface are generated dynamically by a series of Servlets and JavaServer Pages (JSPs), and are presented to the user using HTML forms viewable through any modern web browser.

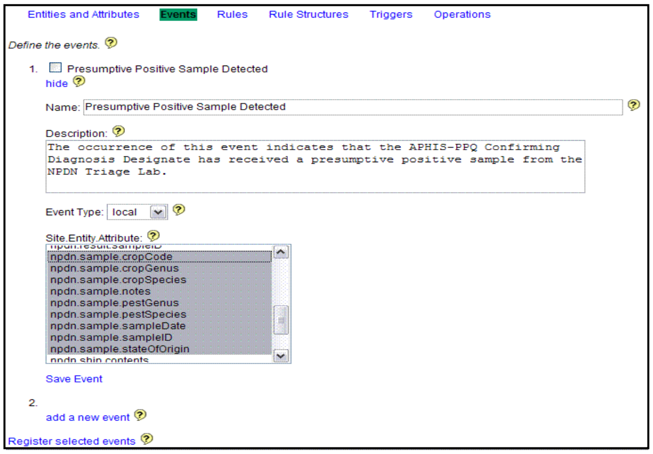

Figure 4. Event Definition

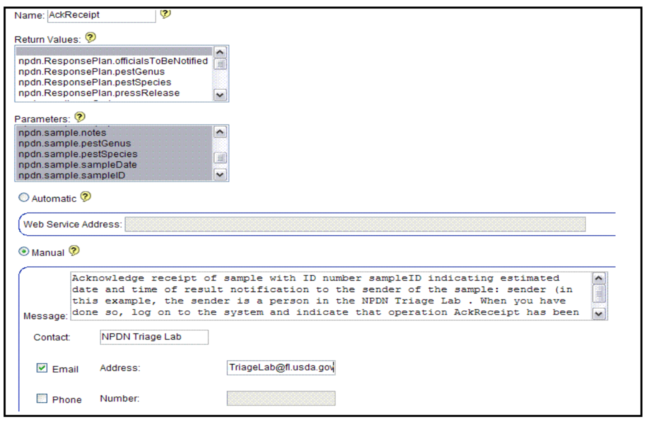

Figure 5. Operation Definition

Each organization provides an **export schema** that describes the data entities and attributes it is willing to share with others. All export schemas of collaborating organizations are integrated to form a **global schema**, which defines all sharable data entities and their attributes in a collaborative federation. The global schema is created and stored at the host site and is downloaded to all collaborating organizations' sites for use in their definitions of events, rules, and application operations. That is, data entities and attributes referenced by them are selected from those defined in the global schema. Typically, the entities and attributes referenced in an event definition would be those whose data (i.e., the event data) can be provided by the site at which the event occurs. The global schema can be rather large. All the data entities and attributes defined in it may not be required when defining an event, rule or operation. To improve the functionality of the user interface, a separate tab on the screen displays a tree structure of the data entities and attributes. The user can select those that are relevant to the definition of a particular event/rule/operation. Only the selected entities/attributes are visible on the event/rule/operation definition screens.

The **event definition** screen is shown in Figure 4. Each event is uniquely identified by its name. A description for each event can be provided during the definition. The event can be categorized as local or global depending on whether

Figure 6. Rule Definition

the organization wants to treat the event data as private or shared respectively. Occurrence of a global event will cause the event data to be sent out to all subscribing organizations (i.e., explicit subscribers) and to those that have applicable rules (i.e., implicit subscribers). The event registration feature (seen at the bottom left of the figure) allows the user to select one or more global events and upload them to the host; this allows all organizations to explicitly subscribe to these events and be notified of their occurrences.

The **operation definition** screen is shown in Figure 5. Each operation has certain characteristics such as its name and input and output parameters. An operation can be automated or manual. An automated operation is made available to the system via a published web service; the URL of the WSDL document that describes the service is all that is needed during definition. The definition of a manual operation specifies the contact person(s), the email address(es) and/or cell phone number(s), along with a message to instruct the person(s) on what operation to perform and how to inform the system after the operation has been performed.

The **rule definition** screen is shown in Figure 6. Each rule has a name, a description, and a state. The state of a rule can be **active** or **suspended**. Each rule belongs to one of the three rule types mentioned in Section 1. Depending on the rule type, the interface dynamically generates an HTML form to allow the user to enter the parameters specific to that type. Some rules may have complicated expressions. The user interface provides the facility to define, display and hide components of these expressions. Defined rules are automatically translated into program code and wrapped as web services. The meta information of the web service for an operation/rule is registered in a service registry at the host to make them available for collaborating organizations.

The user interface also has the capability of defining a rule structure by creating a diagram that specifies the relationships among rules. The diagramming feature gives the user the ability to select and place multiple rules, and then specify connections among them. A connection can either be sequential, parallel (i.e., split construct) or synchronized (i.e., join construct). In the latter case, multiple rules or rule sequences (i.e., the predecessors) converge to a single rule (the successor). The join can be an and-join or an or-join as described in Section 3.

Once a set of events, rules and rule structures has been defined, it is necessary to tie distributed events to distributed rules and rule structures. To this end, the **trigger definition** screen gives the user the opportunity to specify which event(s) will trigger the processing of which rule or rule structure. A drop-down box lets the user select one or more pre-defined events, and a preview window shows the description and the entities and attributes associated with each event. Following the selection of event(s), the user can select a predefined rule/rule structure using a preview pane.

At the time of writing this paper, a proposal for a system demonstration has been submitted to the System Demonstration session of the 8th Annual International Conference on Digital Government Research. If accepted, the prototype will be demonstrated at the conference.

6. CONCLUSION

In this paper, we presented our idea of capturing multi-faceted human and organizational knowledge by using three popular types of knowledge rules and rule structures. The architecture of our system and the user interface were described. We also introduced the technique of managing dynamic event data and processing distributed and heterogeneous rules to achieve knowledge sharing by converting rules and rule structures into web services for their uniform discovery, invocation and interoperation in a web service infrastructure. The processing technique was explained using NPDN's SOP as an example.

There are several research issues that are being investigated, one of which is security and trust policies. Collaborating organizations need to negotiate and establish the policies to be enforced by the knowledge sharing system. We are interested in specifying these policies in knowledge rules and rules structures so that they can be processed uniformly with other knowledge rules. The second issue is about ontology. Since export schemas are defined by different organizations, the terms used to name data entities and attributes may have semantic discrepancies. Manually mapping terms used by different organizations to form the global schema can be rather tedious and error-prone. We are investigating the use of a domain ontology managed by an ontology manager [3] to either automatically or semi-automatically deal with ontological mappings by reasoning on the underlying concepts of terms. Another issue is that of scalability. When a large number of collaborating organizations wish to interact with each other, the number of events, rules, rule structures, triggers and operations can become prohibitively large. Having just one host site to manage registered meta information can be a performance bottleneck. One approach of mitigating this is to use multiple host sites, each dealing with a categorized subset of events and rules that are pertinent to the problems that organizations want to solve collaboratively.

8. ACKNOWLEDGMENTS

This projected is supported by NSF grant no. IIS-0534065.

7. REFERENCES

[1] Animal Plant and Health Inspection Service, http://www.aphis.usda.gov.

[2] Nick Bassiliades, Ioannis Vlahavas, I., and Ahmed Elmagarmid, "E-DEVICE: An Extensible Active Knowledge Base System with Multiple Rule Type Support," *IEEE Transactions on Knowledge and Data Engineering, 2, 5, 824-844*, Sep 2000.

[3] Howard Beck, *On-line Content Development Tools*, http://orb.at.ufl.edu/ObjectEditor, 2005.

[4] Business Rules Group, "*Defining Business Rules – What Are They Really?*," Final report, http://www.businessrulesgroup .org/first_paper/BRG-whatisBR_3ed.pdf, 2000.

[5] Seema Degwekar et al., "Event-triggered Data and Knowledge Sharing among Collaborating Government Organizations," Accepted for Publication, *8th Annual International Conference on Digital Government Research*, May 20-23, 2007.

[6] National Plant Diagnostic Network, http://www.npdn.org.

[7] The National Plant Diagnostic Network Standard Operating Procedure for APHIS-PPQ Pest of Concern Scenario – General SOP, 2005.

[8] Florian Rosenberg and Schahram Dustdar, "Towards a Distributed Service-Oriented Business Rules System," *IEEE Third European Conference on Web Services Proceedings*, Nov 14-16, 2005.

[9] Isabelle Rouvellou et al, "Combining Different Business Rules Technologies: A Rationalization," *OOPSLA 2000 Workshop on Best-practices in Business Rule Design and Implementation Proceedings*, Oct 15, 2000.

[10] John Sowa, *Knowledge Representation: Logical, Philosophical and Computational Foundations*, Pacific Grove, CA: Brooks Cole, 2000.

[11] Jeffrey Ullman. *Principles of Database Systems, 2nd Ed,* Rockville, MD: Computer Science Press, 1982.

[12] Jeffrey Ullman. *Principles of Database and Knowledge-Base Systems,* Rockville, MD: Computer Science Press, 1988.

[13] U.S. Congress. Office of Technology Assessment, *Harmful Non-Indigenous Species in the United States*, OTA-F-565 Washington, DC: U.S. Government Printing Office, 1993.

[14] Jennifer Widom and Stefano Ceri. *Active Database Systems, Triggers and Rules for Advanced Database Processing*, San Mateo, CA: Morgan Kaufmann, 1996.