Distributed Processing of Event Data and Multi-faceted Knowledge in a Collaboration Federation

Stanley Y.W. Su, Howard Beck, Seema Degwekar, Jeff DePree, Xuelian Xiao, Chen Zhou, Minsoo Lee

Abstract— **This paper presents the goal, accomplishments and research issues of an NSF project. The project aims to develop a distributed event-triggered knowledge sharing network (ETKnet) for government organizations to share, not only data and application operations, but also knowledge embedded in organizational and inter-organizational policies, regulations, data and security constraints as well as collaborative processes and operating procedures. A unified knowledge and process specification language has been developed to formally specify multi-faceted human and organizational knowledge in terms of three types of knowledge rules and rule structures. A user-friendly interface is provided for collaborating organizations to define events of interest as well as application operations, knowledge rules, rule structures and triggers. Events are published in a global registry for browsing, querying, event subscription and notification. Rules and rule structures are automatically translated into Web services for distributed processing in ETKnet. Event data are dots that can be connected dynamically across organizational boundaries through the interoperation of knowledge rules, processes and application operations.**

Index Terms— **knowledge Management, Distributed System, Collaboration Technology, Event-based System.**

I. Introduction

GOVERNMENT organizations world-wide are facing complex problems such as illegal immigration, terrorism, disease outbreaks and natural disasters. Effective resource sharing, collaboration and coordination among government organizations are needed to solve these and other complex problems. They need to share, not only data and application system functions, 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 and inter-organizational policies, regulations, constraints, processes and operating procedures. A common means of representing knowledge is to use knowledge rules [18]. Three types of knowledge rules have been found to be useful in many applications [1], [16]: integrity constraints [22], logic-based derivation rules [23], and action-oriented rules [3], [25]. 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. Furthermore, based on our work in both e-business and e-government applications [5], [7], we have found that it is necessary to use a structure of these different types of rules to model a complex collaborative policy or process. Also, a collaborative process and operating procedure may need to be enacted when some important event occurs. It will be very useful if knowledge rules and processes/procedures can be specified in a unified specification language so that they can interoperate. By interoperability, we mean that rules of different types and processes/procedures can interface with one another seamlessly. For example, when an event occurs, there is data associated with the event (i.e., event data). A logic-based derivation rule defined by one organization may deduce from the event data some additional data for use by an action-oriented rule of another organization. This action-oriented rule may activate a structure of operations, which produce some data that need to be verified by a constraint rule of yet another organization. Or, the above rules and operations can be specified by a single organization in a rule structure triggered by the event.

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

Manuscript received January 15, 2008. This work is supported by NSF under Grant IIS-0534065.

Stanley Y.W. Su (corresponding author), Seema Degwekar, Jeff DePree and Xuelian Xiao are with Database System R&D Center, Department of Computer Information Science and Engineering, University of Florida, Gainesville, FL 32611 USA (phone: 1-352-392-2693; e-mail: {su,spd,jdepree,xxiao}@ cise.ufl.edu).

Howard Beck and Chen Zhou are with Department of Agricultural and Biological Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: {hwb, czhou}@ufl.edu).

Minsoo Lee is with Department of Computer Science and Engineering, Ewha Womans University, Seoul 120-750 Korea (e-mail: mlee@ewha.ac.kr).

entirety for access by other organizations is not an option. Apart from the security reason, 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 (i.e., event data) and process only those knowledge rules and processes/procedures that are "applicable" to the event data. A rule, process or procedure is applicable to an event, if the objects and attributes specified in the event data form a superset of the input data needed for processing the rule, process or procedure. Initially, the event data set contains only the data associated directly with the event occurrence. However, as relevant rules and operations are applied on the event data, new data can be generated and old data can be modified, thus making some other distributed knowledge rules and rule structures applicable. Multiple rounds of event data transmission, rule processing, and data aggregation may take place in order to produce all the data that are 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, rule structures and processes/procedures would be ideal for any collaborative federation. It is therefore the goal of our project to research and develop such a system.

An important issue 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 [15], and provide a common interface to enable these rule engines to exchange the data generated by rules. In our opinion, this approach is not ideal either because it will result in a very complex system that is difficult to operate and maintain. Another issue is the specification and enactment of collaborative processes and operating procedures. Traditionally, business processes and workflow processes are defined by some definition languages such as BPEL [13] and WPDL [26], and processed by some business process execution engine or workflow management system. Typically, these engines and systems do not interoperate with knowledge management systems.

This paper reports the main extensions made to our earlier system reported in [6], [7]. We have extended our knowledge specification language and user interface to allow integrated specification of knowledge rules and processes/procedures. Operation structures which model processes or operating procedures can be specified in the action or alternative action part of a condition-action-alternative-action rule. Defined rules (including processes/procedures) and rule structures are then automatically translated into code and wrapped as web services for their processing and interoperation in a web service infrastructure. This approach avoids the use of multiple rule engines and workflow/process management systems, and the problem of rule-to-rule conversions. Additionally, we have added an ontology management system to the system architecture of ETKnet.

II. National Plant Diagnostic Network

The collaborative federation that serves as our application domain is the National Plant Diagnostic Network (NPDN [11]). 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 [24]. NPDN has developed a general standard operating procedure (SOP [21]) to combat a pest-of-concern situation, which details the steps to be taken when such a bio-security event takes place. At present, NPDN organizations are developing their regional operating procedures using the general SOP as the guide. These procedures are, in most cases, carried out manually. There is a need for their automation.

III. Architecture of ETKnet

ETKnet has a peer-to-peer server architecture as shown in Fig. 1. All collaborating organizations have identical subsystems installed at their sites. Each site creates and manages its own events, rules, rules structures, triggers and operations. Their specifications are registered at the host site of a federation.

A user interface is used at each site to define sharable events, rules, rule structures, triggers and application system operations. The system supports three types of rules: logic-based derivation rule adopted from RuleML [17], constraint rule patterned after [14], and action-oriented rule [25]. Instead of using event-condition-action (ECA) rules, our action-oriented rules have the format: condition-action-alternative-action (CAA). We separate event (E) from CAA rules because they can be defined by different collaborating organizations independently. In both action and alternative action clauses, a single manual or automated operation can be specified.

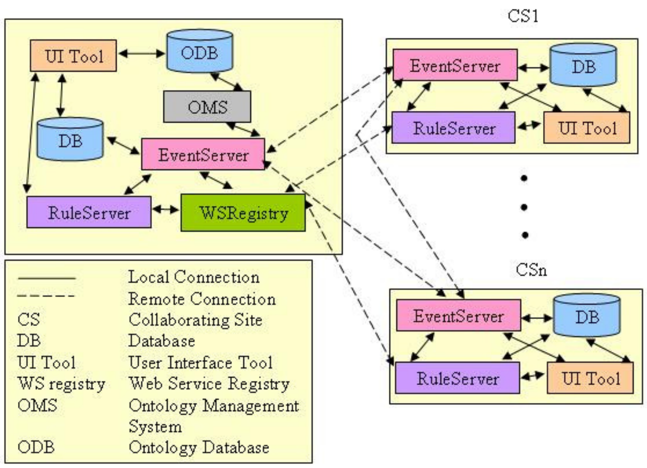

Fig. 1. System Architecture

Alternatively, a structure of manual and/or automated operations can be used to model a complex process or operating procedure. We adopt the structural constructs found in BPEL and WPDL; namely, sequential, switched, unordered, selective, repeated, And/OR/XOR split, and AND/OR join.

A *rule structure* is 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, …s_n, states that after executing rule r, rules s_1, s_2, … s_n can begin their execution in parallel. An *and-join* relationship between a collection of rules r_1, r_2, …, r_n and rule s states that rule s may begin its execution only after r_1, r_2, …, r_n complete theirs. An *or-join* relationship between a collection of rules r_1, r_2, …, r_n and rule s states that rule s may begin its execution after a specified number of rules in the collection complete theirs. We note here that the rule structure is simpler than the operation structure because each rule represents a higher-level specification than the primitive operations. All we need is the capability to specify sequential, parallel and synchronization of rules in rule processing.

We have designed and implemented an XML-based knowledge and process specification language. Interested readers can visit our website for the schema of the language: http://www.cise.ufl.edu/research/ETKnet/RuleBase.xsd

When a shared rule or a rule structure is defined, 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 WSRegistry at the host site. The generated rule code is installed at the site that defines the rule or rule structure for both security and efficiency reasons because it typically invokes some local application operations. The integrated specification and processing of knowledge rules and processes/procedures as web services is significant because it enables their interoperation.

The Event Server is responsible for storing information about events defined at that particular site and information about event subscribers. Events defined at all sites are registered with the host site. Users of all organizations can browse or query the registry to subscribe to the event. They can also define triggers to link events to rules and rule structures. In both cases, they are "explicit subscribers". Triggers can be automatically and dynamically generated by the system for those sites that contain applicable rules and rule structures. In that case, the organization that defines the rule or rule structure is an "implicit subscriber" of the event. The subscriber information is automatically downloaded to the Event Server of the site that defines the event for event notification purposes. When an event occurs at a particular site, its Event Server serves as the coordinator of a knowledge sharing session. It carries out event notification by sending event data to both explicit and implicit subscribers to trigger the processing of rules. The Event Server also handles the aggregation of new event data returned from collaborating sites as the result of rule processing. The new version of the event data may trigger another round of rule processing. Multiple rounds of event data transmission and rule processing may take place until no applicable rules exist in the distributed system. At that time, all relevant organizations would have received all the data that are pertinent to the event occurrence.

In a collaborative environment, terms used by different organizations to define events, rules, triggers and operations can be quite different. It would be beneficial to define a domain-specific ontology to resolve the discrepancies and identify the similarities among specified terms and to facilitate search. Thus, the host site also has an ontology management system including an ontology database. An ontology is a representation of knowledge in a specific domain. It defines a set of concepts within this domain and their relationships. We have defined a domain-specific ontology based on NPDN's standard operating procedure. The ontology database contains the concepts and relationships associated with the event data, terms used in rule and operation specifications, and roles that different people play in NPDN organizations.

The ontology management system we have developed is called Lyra [4]. It consists of a set of tools to create, store and maintain an ontology. The logic structure of Lyra consists of three layers. The bottom layer is a persistent layer, which stores the ontology as classes, individuals, and properties. The middle layer is an inference layer, which reads the concepts and individuals from the persistent storage, converts them into a normalized form, and performs ontology reasoning which is used for query processing. We enhanced the description logic reasoner, Pellet [20], and use it to perform inference tasks. The top layer is a term layer, which has a similar structure as the WordNet [12] lexical database. This layer maps the terms used in users' queries as well as terms used in the specifications of events, rules, and operations to the objects defined in the ontology, so that the discrepancies and similarities between the specified terms can be resolved and identified. We have also developed an ontology editing tool for creating and editing terms and concepts, a rule matcher to facilitate the user's querying for sharable rules, and an application interface for the Rule Server and the Event Server to query for people, who play specific roles in NPDN organizations.

IV. Distributed Event and Rule Processing

In this section, we use the application in the NPDN domain to explain our distributed event and rule processing technique. We pick a subset of the communication path outlined in the general SOP mentioned in Section II and use it as an example.

Fig. 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. 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. These labs have applicable rules that can make use of the event data to provide some more relevant information. APHIS performs preliminary diagnosis on the sample (step 3). NPDN Regional Hub Lab informs the appropriate personnel of the sample status (also step 3). These procedures at NPDN Triage Lab and NPDN Regional Hub Lab are modeled using heterogeneous rules and rule structures. Details of these procedures are given later in this section.

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 there are applicable rules. Thus, multiple rounds of event data transmission, rule processing and data aggregation can take place as event data is dynamically changed by rules.

Fig. 3 describes the rule structure executed at NPDN Triage Lab upon receiving a sample. This is a structure with both link and split relationships. The first rule, NTLR1, is concerned with acknowledging the receipt of the sample to the sample submitting entity, which can be another diagnostic lab or an independent grower. It asks the lab diagnostician to perform a preliminary diagnosis on the sample. Rule NTLR2 checks if the lab has a web or distance diagnosis capability. If so, NPDN Hub Lab is contacted to perform the distance diagnosis. Otherwise, a photograph of the sample is emailed. Rule NTLR3 instructs the staff on how to divide and ship the sample to the other labs for a confirming diagnosis. If it is a routine sample, or the Hub Lab has provisional approval to perform the confirming diagnosis, the sample is to be sent to NPDN Hub Lab; otherwise it is to be sent to a confirming diagnosis designate. Rule NTLR4 instructs the staff to contact the campus safety officer to apprise him/her of a presumptive positive sample in the system.

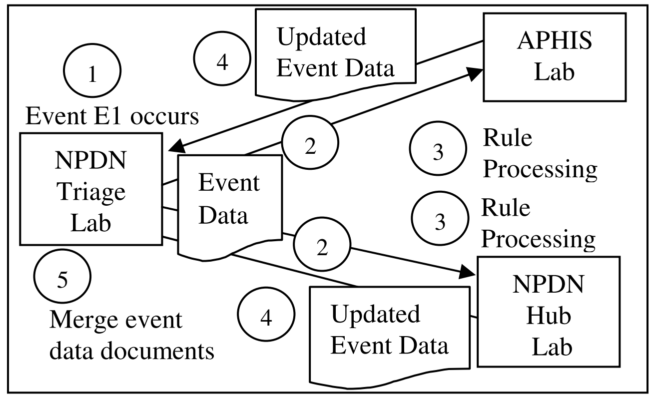

Fig.2. Distributed Event and Rule Processing

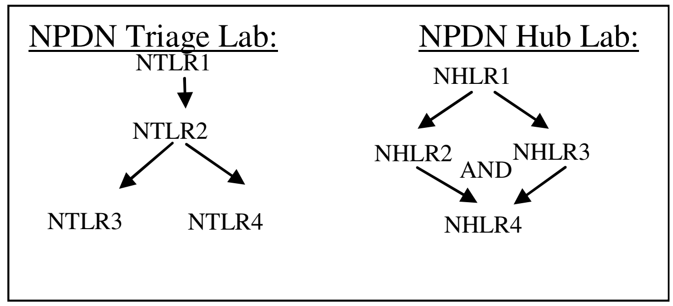

Fig.3. Rule Structures (NPDN Triage Lab, NPDN Hub labs)

Fig. 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 the result back to Triage Lab. The first rule, NHLR1, acknowledges the receipt of the presumptive positive sample by NPDN Triage Lab. It also checks to see if proper sample shipping procedures were followed and informs the administrator if otherwise. Rule NHLR2 is concerned with asking a local expert to perform some preliminary diagnosis on the sample. Rule NHLR3, which is processed in parallel with NHLR2, contacts other personnel such as a State Plant Regulatory Official, and a State Plant Health Director to inform them of the presumptive positive sample in the system. It is also concerned with determining whether or not Regional Hub Lab needs to send the sample to APHIS Lab for further confirmations. After receiving the diagnosis from the local expert, and if APHIS Lab requires Regional Hub Lab to send the sample, rule NHLR4 is concerned with ensuring that the sample is sent. For space reasons, we do not include the rules and rule structures at APHIS Lab.

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 ETKnet 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 later. Through our user interface, the agent can report to the system that the operation has been done and provide its results.

V. Implementation

All components shown in Fig. 1 have been implemented. Different from the earlier version that was demonstrated [8], the current system supports the integrated specification and processing of rules and processes/procedures using the extended specification language, user interface and ETKnet. The algorithms used for translating rules and rules structures into web services, for aggregating event data to produce a new version of event data, and for avoiding cyclic processing of distributed rules can be found in [7].

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 system. A user interface has been developed to allow collaborating organizations to define their events, rules (including operation structures), rule structures, triggers, and manual and automated operations used in rules. The extended interface enables a user to define an operation structure having various structural constructs in the action part and the alternative action part of an action-oriented rule by creating a diagram to specify the structural relationships among operations. 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. More details about the user interface can be found in [8].

VI. Research Issues

There are several research issues that are being investigated. One is regarding security and trust policies. Collaborating organizations need to negotiate and establish the policies to be enforced by ETKnet. We are interested in specifying these policies in knowledge rules and rules structures so that they can be processed uniformly with other knowledge rules. There are questions concerning who can define and publish events, who can define and modify rules, who can define and modify triggers that link distributed events to distributed rules, and what should be done with those triggers wherein the event(s) specified have been deleted (i.e., considered by the collaboration federation as no longer important). Role-based access control strategy [19], the concept of membership [9], certification-based authentication [10], and PKI technologies are applicable in a federated environment. We shall adopt available technologies and, in some case, extend them for access control and management of distributed events, rules and triggers.

The second issue regards ontology, which was discussed in Section III. Terms used by one organization in its specifications of events, rules, triggers and operations, in its names for entities and attributes of event data, and in its metadata descriptions can be quite different from those used by another organization. Furthermore, different organizations may create similar but slightly different conceptual structures to represent the same concept. People searching for registered events and web services that implement rules and rule structures will likewise face with a mismatch between the terms used in their searches and the terms used to register these data and knowledge resources. In addition to the ontology defined for NPDN's SOP, we are defining a domain ontology for plant disease and pest diagnostics to be managed by the ontology manage system, which will either automatically or semi-automatically deal with ontological mappings by reasoning on the underlying concepts of terms. We are also extending the web services technology by developing an ontology-enhanced web service registry to enable the registration, semantic discovery and invocation of application operations, rules and rule structures.

Another issue is that of scalability. The architecture described in Section III allows organizations to join a federation by installing microcomputers that contains the developed software and tools at their network sites. The network is highly expandable and scalable because more computational power is added as more sharable knowledge is added to the federation. Also, since the components shown in Fig. 1 are implemented as servers, multiple federations can be accommodated: i.e., an organization can be a member of multiple federations and its servers can process different event data sets concurrently in multiple threads. However, as the number of organizations in a federation increases, the number of federations grows, and the number of event occurrences of different event types increases, the performance of the entire network can deteriorate if a centralized host is used. Event types may have to be categorized and managed by multiple hosts. Also, there are data scalability issues. For example, should an organization archive the event data sets associated with all or some of the events it subscribes? Should the distributed network archive all or some of the event data sets? How does the system determine which historical event data to keep? Can the event data associated with an event occurrence grow to a very large size because an event may trigger rules, which in turn post other events to trigger other rules? We are seeking answers to the above questions.

VII. Conclusion

In this paper, we have presented our idea of capturing multi-faceted human and organizational knowledge by using three popular types of knowledge rules and rule structures. Operation structures that model collaborative processes and operating procedures are embedded in action-oriented rules, thus allowing processes/procedures and knowledge rules to be defined in a unified specification language and be processed in an integrated and interoperable fashion. The architecture of ETKnet and the user interface were described with some implementation details. 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. The intended contributions of this paper are: 1) introducing a multi-faceted knowledge representation for capturing organizational and inter-organizational policies, regulations, constraints, processes and operating procedures in an integrated manner, and for sharing distributed heterogeneous rules and rule structures, 2) introducing an effective mechanism for translating high-level knowledge specifications into code and wrapping them as web services for their uniform processing and interoperation in a web service infrastructure, thus avoiding the use of multiple rule engines and a workflow process management system, 3) presenting the architecture of an event-triggered knowledge network, and the distributed event and rule processing strategy by using agricultural homeland security as the application domain, and 4) identifying research issues for our and others' further investigation.

References

[1] Business Rules Group, "Defining business rules - what are they really?" Available: http://www.businessrulesgroup.org/first_paper/br01c0.htm

[2] Bassiliades, N., Vlahavas, I., and Elmagarmid, A. K., "E-DEVICE: an extensible active knowledge base system with multiple rule type support," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 12, pp. 824-844, 2000.

[3] Bry, F., Eckert, M., Pătrânjan, P., and Romanenko, I., "Realizing business processes with ECA rules: benefits, challenges, limits," *Proceedings of Int. Workshop on Principles and Practice of Semantic Web Reasoning*, pp. 48-62, 2006.

[4] Beck, H., Lyra – On-Line Content Development Tools, Available: http://orb.at.ufl.edu/ObjectEditor/

[5] Degwekar, S., and Su, S. Y. W., "Knowledge Sharing in a Collaborative Business Environment", *Proceedings of the Fifth Workshop on e-Business*, Milwaukee, Wisconsin. Abstract on page 60 and paper on CD, 2006

[6] Degwekar, S., DePree, J., Beck, H., and Su, S. Y. W., "A Distributed Event-triggered Knowledge Sharing System for Agricultural Homeland Security", *the Proceedings of the 2007 IEEE Conference on Technologies for Homeland Security*, Woburn, MA., pp.180—185, 2007

[7] Degwekar, S., DePree, J., Beck, H., Thomas, C. S., and Su, S. Y. W., "Event-triggered Data and Knowledge Sharing among Collaborating Government Organizations", *the Proceedings of the 8th Annual International Conference on Digital Government Research*, Philadelphia, PA., pp. 102—111, 2007

[8] Degwekar, S., DePree, J., Su, S. Y. W., and Beck, H., "A Distributed Event-triggered Knowledge Sharing System: System Demo". Received the Best System Demo Award at the 8th Annual International Conference on Digital Government Research, Philadelphia, PA, 2007

[9] Hayton, R. J., Bacon, J. M., and Moody, K., "Access control in an open distributed environment," *Proceedings of the IEEE Symposium on Security and Privacy*, California, pp.3--14, 1998.

[10] Johnston, W., Mudumbai, S., and Thompson, M., "Authorization and attribute certificates for widely distributed access control," *Proceedings of the IEEE Seventh International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp.340--345, 1998.

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

[12] Miller, G. A., WordNet, a lexical database for the English language, Available: http://wordnet.princeton.edu/

[13] OASIS, Business Process Execution Language, Available: http://www.oasis-open.org/specs/index.php

[14] Object Management Group. (2001) Object Constraint Language Specification. Available: http://www.omg.org.

[15] Rosenberg, F., and Dustdar, S., "Towards a distributed service-oriented business rules system," *IEEE Third European Conference on Web Services Proceedings*, Nov 14-16. 2005.

[16] Rouvellou, I., et.al, "Combining different business rules technologies: a rationalization," *OOPSLA 2000 Workshop on Best-practices in Business Rule Design and Implementation Proceedings*, 2000.

[17] Rule Markup Language, Initiative, Available: http://www.ruleml.org

[18] Sowa, J. F., *Knowledge Representation: Logical, Philosophical and Computational Foundations*, Brooks Cole Publishing Co., Pacific Grove, CA, 2000.

[19] Sandhu, R., Ferraiolo, D., and Kuhn, R., "The NIST model for role-based access control: towards a unified standard," *Proceedings of the First ACM Workshop on Role-Based Access Control*, pp.47—63, Berlin, Germany, 2000.

[20] Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y., "Pellet: a practical owl-dl reasoner", *the 3rd International Semantic Web Conference (ISWC2004)*, Hiroshima, Japan, November 2004

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

[22] Ullman, J., *Principles of Database Systems*, 2nd Ed. Computer Science Press, Rockville, MD, 1982.

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

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

[25] Widom, J., and Ceri, S., *Active Database Systems, Triggers and Rules for Advanced Database Processing*, Morgan Kaufmann, San Mateo, CA. 1996.

[26] WFMC, Workflow Management Coalition Interface 1: Process Definition Interchange Process Model, Available: http://www.wfmc.org/standards/docs/if19807m.pdf