Carnegie Mellon University
Research Showcase @ CMU

Institute for Software Research | School of Computer Science

2004

A Prototype System for Transnational Information Sharing and Process Coordination

Violetta Cavalli-Sforza
Carnegie Mellon University

Jaime G. Carbonell
Carnegie Mellon University, jgc@cs.cmu.edu

Peter J. Jensen
Carnegie Mellon University

Follow this and additional works at: http://repository.cmu.edu/isr

Published In
.

This Working Paper is brought to you for free and open access by the School of Computer Science at Research Showcase @ CMU. It has been accepted for inclusion in Institute for Software Research by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

A Prototype System for Transnational Information Sharing and Process Coordination

S. Su, J. Fortes, T. R. Kasad, M. Patil, A. Matsunaga, and M. Tsugawa, University of Florida
V. Cavalli-Sforza, J. Carbonell, and P. Jansen, Carnegie Mellon University
W. Ward and R. Cole, University of Colorado
D. Towsley and W. Chen, University of Massachusetts
A. I. Antón and Q. He, North Carolina State University
C. McSweeney, University of Belize
L. de Brens, J. Ventura, and P. Taveras, Pontificia Universidad Católica Madre y Maestra
R. Connolly, C. Ortega, and B. Piñeres, Organization of American States
O. Brooks, National Drug Abuse Control Council, Belize
M. Herrera, National Drug Council, Dominican Republic

Abstract

Global problems such as disease detection and control, terrorism, immigration and border control, illicit drug trafficking, etc. require information sharing, coordination and collaboration among government agencies within a country and across national boundaries. This paper presents a prototype of a transnational information system which aims at achieving information sharing, process coordination and enforcement of policies, constraints, regulations, and security and privacy rules by integrating a distributed query processor with form-based and conversational user interfaces, a language translation system, an event server for event filtering and notification, and an event-trigger-rule server. The Web-services infrastructure is used to achieve the interoperation of these heterogeneous component systems.

1. Introduction

All countries in the world are facing global problems such as disease detection and control, terrorism, immigration and border control, illicit drug trafficking, and others. The solutions to these complex problems requires, among other things, information sharing, close communication, coordination and collaboration among government agencies in many countries. There is an urgent need for developing and integrating advanced information technologies to enable government agencies to share information and work together within a country as well as across national boundaries. Solving these complex problems in the transnational setting presents many new technological challenges. We list some of them below and point out the technologies needed to meet these challenges:

1) Data heterogeneity: Data collected by different agencies in different countries have structural and semantic differences. An integrated, global conceptual schema needs to be constructed by integrating the heterogeneous data to provide users with a uniform view of the distributed data, and a distributed query processing system is needed to access and manipulate the data. The global conceptual schema should be presented in the different natural languages used by users.

2) Language heterogeneity: Collaborating countries may use different natural languages. Data and documents in an unfamiliar language cannot be readily used by agencies in other countries. Automated natural language translation is needed to achieve information sharing.

3) Heterogeneity in people and working environments: Government agency employees have varying proficiency in the use of information technologies; their requirements for interaction with these systems can be quite different. A choice of user-friendly interfaces is needed to suit a spectrum of users. Working environments can also be quite different. In some there is full access to the Internet and other computing facilities, whereas in others, particularly in developing countries, Internet access may be unreliable or altogether lacking. It is important for a transnational information system to provide different means of communication (e.g., communication by emails, short messages via cell phones, paging, telephone, etc.)

4) Heterogeneity in government policies, regulations, constraints, and security and privacy rules: The provision of local autonomy is of paramount importance to foster international cooperation and collaboration because each country may have its own policies, regulations, constraints and rules regarding

what information can be accessed by whom, and when and how information can be used. These policies, regulations, etc. may change with time, so they definitely should not be hard-coded into a transnational information system. A powerful, high-level rule specification language and an efficient rule processing system are needed to define and enforce them, respectively.

5) Difficulties in inter-agency and inter-government communication and coordination: At the present time, agencies within a country often have great difficulty in keeping close communication and coordination amongst themselves. Communication and coordination among collaborating countries are even more difficult. Collaborating countries can benefit from each other by keeping one another informed of important events (e.g., the outbreak of a disease, a terrorist's movements, etc.), by automatically delivering relevant information, and by activating operations and processes in response to the occurrences of events. Tools and mechanisms for supporting event publication, subscription, filtering and notification, and for performing event and rule-based triggering of operations and processes are needed.

6) Heterogeneity in computing platforms: Different government agencies worldwide use dissimilar hardware, software, operating systems, data management systems, and application systems to perform their functions. There is a need for a common, standard-based infrastructure for accessing and inter-operating these resources over a wide-area network like the Internet.

Supported by a grant from the National Science Foundation of the United States, researchers at seven universities (Carnegie Mellon University, University of Belize, University of Colorado, University of Florida, North Carolina State University, University of Massachusetts and Pontificia Universidad Católica Madre y Maestra of the Dominican Republic) and experts from agencies in three countries (the Organization of American States (OAS) of the United States, the National Drug Abuse Control Council of Belize's Ministry of Health, and the National Drug Council of the Dominican Republic) are developing information technologies to enable resource sharing, coordination and collaboration among agencies of the collaborating countries. The technologies and component systems can be replicated and installed at different sites to form an integrated transnational information system and enable the establishment of collaboration grids over the Internet. In each collaboration grid, participant countries can seamlessly integrate transnational activities and policies into their national government processes so that each country's public service agencies can have controlled access to and use of their own data and application resources as well as the data and resources of agencies in collaborating countries. The agencies of each country can participate in the activities of multiple collaboration grids to share information and application resources needed to solve different transnational problems.

This paper presents the prototype of a transnational information system that integrates a distributed query processing system, an event server, and an event-trigger-rule server developed at the University of Florida, a machine translation system developed by Carnegie Mellon University, a conversational interface developed at the University of Colorado, and a Web-services infrastructure jointly implemented by collaborating universities to achieve the interoperability of these component systems. In order to test and demonstrate the developed technologies, the project's initial focus is on information sharing and process coordination as related to border control against illegal immigration and drug trafficking. While the current pilot project focuses on connecting border posts within and between Belize and the Dominican Republic, the technologies can be used in other problem domains to enhance international cooperation and more effective government in general. The design of the prototype system is based on a requirements gathering and analysis carried out by team members at the North Carolina State University. The prototype system has been implemented and demonstrated at the ministerial level in Belize in December 2003. It will also be demonstrated at dg.o2004 [15]. A second demonstration of a more advanced prototype is planned to take place in the Dominican Republic during 2004.

2. System Architecture

The architecture of the prototype system is shown in Figure 1. It consists of a host site and the sites of participating countries and their agencies. In the prototype implementation, we use two sites (one in Belize and the other in the Dominican Republic) to represent the countries' agencies. The data used to test and demonstrate the developed technologies are artificial data, generated based on the port-of-entry

and exit forms used by these two countries. No real data is used for testing and demonstration purposes for privacy protection reasons.

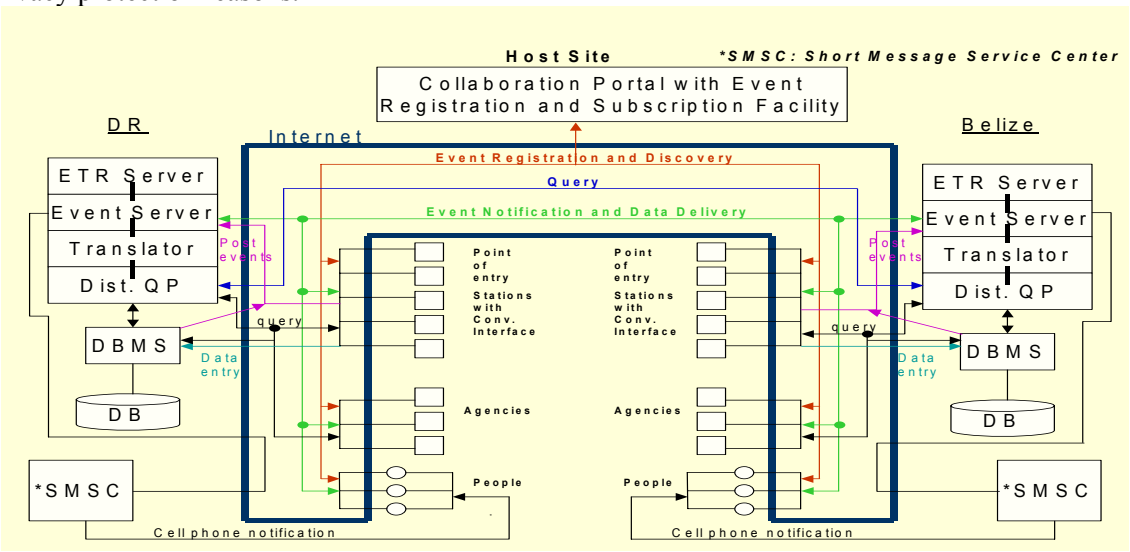

Figure 1. Architecture of the Prototype System

Both countries have local databases that store immigration, border control and government process-related data. These databases are managed by their own local, heterogeneous Database Management Systems (DBMSs). The agents at ports-of-entry in each country use the local DBMS to enter, access and manipulate their data. Those data that a country is willing to share with the agencies of another country are specified in an export schema, and the integration of all the export schemas forms a global schema. The global schema is used to generate query forms in different languages, through which authorized users in participating countries can query against the distributed databases stored in any of these countries. The users are personnel at port-of-entry stations, government agencies and other authorized individuals.

The replicas of software components installed at each of the participant countries' sites are briefly explained below. More details are given in Section 3. The **Distributed Query Processor (DQP)** provides a **Form-based Interface** as well as a **Conversation-based Interface** to allow users to query the distributed data**.** It uses a **Machine Translation System** to translate between English (used in Belize) and Spanish (used in the Dominican Republic) queries and data so that a user can issue a query in one language and receive the query results in the same language, regardless of the language of the data. The Distributed Query Processor with its interfaces and the Machine Translation System enable transnational information sharing. Another main function provided by the prototype system is event-trigger-rule processing. Authorized users in the participating countries can define and register events of common interest (e.g., a person wants to enter a country, or a person is on a watch list) at the Host site by using its **Event Registration and Subscription Facility.** Other users can browse and subscribe to these events and specify event filtering condition(s) (e.g., the person entering the country or the person on the watch list is from a certain country) for receiving event notifications when the subscribed events occur and the filtering conditions are satisfied. The subscribers also specify the desired means of notification (e.g., by emails, short messages to cell phones, and/or activation of application programs or processes defined as Web-services). The subscription and filtering information of an event is sent to the **Event Server** of participating sites where the event may occur. When an event occurs, the Event Server processes the registration and filtering information to decide which subscribers to notify. It will also notify the local **Event-Trigger-Rule (ETR) Server** to trigger rules associated with the event. Additionally, it will also notify the Event Servers of other collaborating countries, which in turn notify their ETR Servers to process the rules associated with the event that occurred. Events are parameterized; the values of the parameters specify the information related to a specific occurrence of an event type (i.e. the event data). The event data can be passed to rules for examining the data conditions associated with the event.

Distributed rules are used to specify different countries' policies, regulations, constraints and security and privacy rules and are enforced by replicas of the ETR Server. The Event Registration and Subscription Facility at the host site and the Event Server and the ETR Server at participants' sites together enable the close communication, coordination and collaboration of participating countries and their agencies.

In this work, we adopt the Web services technology [9] for achieving the interoperation of component systems over the Internet. Shareable operations provided by each system component are defined as Web-service operations. The invocation of these operations is achieved by message passing using the Simple Object Access Protocol (SOAP [1]). The key features of this architecture are: 1) the integration of several information technologies (distributed query processing, event-trigger-rule management, machine translation and conversational interface) to enable information sharing and process coordination across multiple countries, 2) the replication of system components at multiple participant sites to form a peer-to-peer server architecture, which is highly scalable, and 3) the interoperation of system components through the common Web-services infrastructure.

3. Functional Description of Component Systems

3.1 Distributed Query Processor

The DQP, installed at both the Belize and Dominican Republic sites, allows users to query for port-of-entry and exit data stored in the local databases of both countries without having to know their physical locations. It has two components: the global query processing component (GQP) and the local query processing component (LQP). The GQP has a Global Search Form, which is automatically generated based on the integrated global schema derived from a number of export schemas. Each export schema specifies the meta information of the port-of-entry and exit data of a participating country. The export schemas specified in different languages are displayed to a person who is knowledgeable in different languages. Through the user interface, that person can specify the equivalence relationships among data attributes, which are used by the system to construct a set of attribute mapping tables. The constructed tables are downloaded to all participant sites for data mediation purposes. If an export schema is modified or if a new export schema is added, the above process of forming the global schema and constructing the attribute mapping tables is repeated. The global search form is displayed as a Java Server Page in the natural language used for issuing queries. All the users with privileges to use this system have associated user profiles comprising username, password, nationality, and a set of roles from a predefined role set. This user profile information is stored using the role-based authentication system provided by the Tomcat Web Server. A user's role determines the privileges of his/her information access. GQP is deployed as a web application under Tomcat at the two query issuing sites and is thus accessible through the Web. Based on the data distribution information stored in a data dictionary, the GQP decomposes a query into sub-queries and sends them to the LQPs at the participants' sites, from which data are to be retrieved.

In our transnational information system, we adopt the Web services technology for communication among and invocation of its component systems over the Internet. Each replica of the LQP is made accessible over the Internet as a Web-service. The GQP plays the role of a Web-service client and invokes the Web-service of LQPs by wrapping sub-queries in SOAP messages and sending them to LQPs using HTTP. As shown in Figure 2, the LQP at each site has a wrapper that communicates with the local DBMS, the Event Server, the ETR Server and the Machine Translation System. The wrapper performs the following key functions. First, it receives the string sub-query in XML format, parses the XML string using a Document Object Model (DOM)-based parser, and extracts the query and role information from it. Second, it connects to the local Event Server to post a synchronous event (*qaccess*); the event triggers a rule to check the access rights of the query issuer on the attributes being accessed in the query. Third, it deals with the mediation of data heterogeneities [10] between local databases. The global attribute names referenced in the query are mapped to the corresponding local attributes names used in the local database via attribute mapping tables. Since attribute values are stored in different languages, sometimes it becomes necessary to translate attribute values specified in a query in order to query the local database. For example, in the port-of-entry form, an immigration agent uses a comment field to make comments

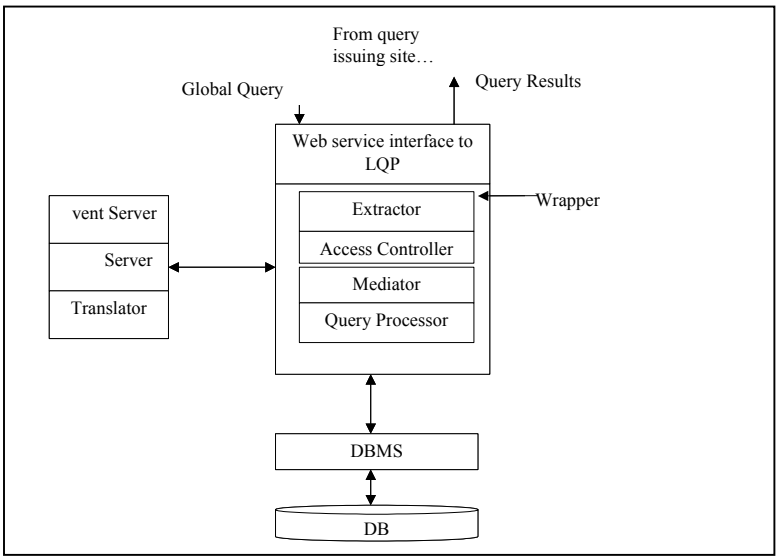

Figure 2. Architecture of the LQP Component

in his/her language about a person entering the country. If a query issuer from a country that uses a different language wants to search the comment field for a certain word or words, the query word(s) must be translated into the language used in the local database before the search can be performed. In this case, the Machine Translation System is called to do the translation. Additionally, after the local database has been queried and before sending the query result back to the GQP, the attribute names in the local query and its result are mapped back to the global query attribute names by a mediator sub-component of the wrapper. The GQP accepts the query results from possibly multiple LQPs and applies suitable Extensible Style Sheet Transformation (XSLT) to convert the results from XML to HTML format and display them in the web browser. The style sheet that is applied to the query results to generate the final display page is also available in all supported natural languages. The style sheet to be applied is determined by the user's country of citizenship information stored as part of his/her profile in the web server's configuration.

3.2. Conversational User Interface

In addition to the form-based interface discussed in the preceding sub-section, the system also provides a conversational interface to the Distributed Query Processor. With this interface, users type natural language queries to the system in the same way that they would request the same information from another person. The conversational interface provides the following functions:

- Takes natural language input, and generates queries to the DQP.
- Receives responses from the DQP and presents the results to the user.
- Maintains dialogue context and resolves references. The user does not have to specify again information that is already in the context and can use pronouns in a natural way.
- Generates clarification requests and error messages. If the user request or the records returned from the database are ambiguous, the system can ask questions to clarify the user's intent. If the system cannot understand the question or detects that the request is beyond the capabilities of the system, it generates a natural language error response to the user.
- Provides a simple help facility to answer questions about the capabilities of the system.

The interface enables query sequences like the following:

list people entering after 3-1-2003 and departing before 6-1-2003
where was laura chang's passport issued
what's her port of entry and date of birth
what about mingo garcia
how many times has he entered the country in the last two years
when is the most recent

The conversational interface was developed using the Semantic Parser and Dialogue Manager modules from the CU Communicator [12]. These modules have been designed to be robust to ill-formed

input and easy to author. User queries are first passed to the Phoenix semantic parser [16], which extracts relevant information into domain specific frames. For example, the query:

list people entering after 3-1-2003 and departing before 6-1-2003

results in the extracted frame:

Arrive_Depart:[Attribute_Select].[Field].name
Arrive_Depart:[Key-Val].dateofentry
[Date_Range].[Gt].[Date].[Month_Num].3 [Day_Num].1 [Year].2003
[Date_Range].[Lt].[Date].[Month_Num].6 [Day_Num].1 [Year].2003

In this example, *Arrive_Depart* is the name of the frame relating to requests for information contained on arrival/departure forms. The individual pieces of information are slots in the frame. The parser provides the capability for grammar rules to map the extracted word strings to a canonical form. In this case, they are mapped onto field names that appear in the database. So even though the terms *name* and *dateofentry* do not appear in the input, they are in the extracted frame.

The output of the parser is passed to the Dialogue Manager (DM), which is responsible for interfacing with the DQP and generating output to the user. The DM also maintains a dialogue context and history to be able to resolve elliptical and anaphoric references. The current extracted frames are merged with the context to produce a new context. For example, after the sequence:

port of embarkation and date of birth for laura chang
what about Mingo Garcia

the context is:

Arrive_Depart:[Atr_Select].[Field].port-of-embarkation-city [Field].dateofbirth
Arrive_Depart:[Key-Val].[First_Name].Mingo [Last_Name].Garcia

After resolving references and generating the current context, the DM generates a request for the DQP. The query consists of the context information for key-value pairs and relations in an XML format. For example:

```
<ATTRNAME>lastname</ATTRNAME>
<OPERATOR>=</OPERATOR>
<ATTRVAL>Garcia</ATTRVAL>
```

The XML query is sent to the DQP via a Java-based interface, and the records returned from the database are received by the same Java routine, also in an XML format. The returned information is parsed from the XML text and presented to the user.

The parser code is completely domain independent and the dialogue manager code is almost domain independent. Developers need to write only a very small amount of code for the DM to implement a new application domain. Domain specific parser information is placed into two ASCII files, the *frames* file, which defines the frames used by the system, and a grammar file, which specifies the semantic grammar patterns that map word strings onto slots in the frames. Domain specific information for the dialogue manager is contained in the *task* file, which specifies the prompts and responses for the system and templates for generating the queries to the DQP.

Users are allowed to use either the form-based interface or the conversational interface. All transactions are logged. Analysis of the log files will allow us to determine how the interfaces are used and user preferences for the interfaces.

3.3 Machine Translation System

The Machine Translation (MT) System is called to translate the contents of the comment field in a database record. We use Carnegie Mellon University's Pangloss-Lite (Panlite) Multi-Engine MT system [7], with recent enhancements [2,3]. Panlite supports using multiple translation engines in parallel.

Given a sentence to translate, each engine provides a translation (along with a score for each translation) for either the full sentence or fragments of the sentence. Translation candidates are placed in a chart as 'edges' covering the input or some portion of it. One component of the system, the language modeler, uses statistical knowledge of the *target* language (the language the system is translating into) to select or piece together from the chart the best scoring translation(s) that cover the entire input.

The Panlite system supports the integration of widely different MT engines, but provides three built-in engines in addition to the language modeler: an Example-Based MT (EBMT) engine, a Glossary engine, and a Dictionary engine. At its simplest, the EBMT translates by matching new input in the *source* language (the language the system is translating from) against source sentences in previously seen examples of source-target sentence pairs.[1] If it cannot find a match for the entire input sentence, it tries to match all possible multi-word input fragments and posts to the chart what it believes to be the corresponding translations. At times, pieces of the input to be translated cannot be matched against any previously seen source sentences, so there will be holes in the translations produced by the EBMT system and it is useful to back off to the Dictionary engine to obtain single-word translations. Finally, a Glossary engine can supplement the translations provided by EBMT with human-supplied translations for phrases. The following translation pairs exemplify the quality of translation we are getting from the system when it has seen similar but not identical source sentences. Usually the output is understandable even if grammar problems are present. Notice that sometimes it is the input that is not fully grammatical.

S: El viajero se presentó en la frontera sin documentos.
T: *Passenger, presented himself at the border without documents.*

S: El viajero no entendía español.
T: *The traveler not understood Spanish.*

S: *Spoke with central office.*
T: Habló con oficina central.

S: *Mingo Garcia traveling from Flores, Peten, crossed at 11:25 a.m.*
T: Garcia Mingo como Flores, Petén, cruzado A las 11 : 25 a.m.

The choice of EBMT as the core translation engine is a choice dictated by the requirements and the constraints of the application. No MT system is currently capable of producing output comparable in quality to that of a good human translator. The best output quality is usually found in Interlingua or Transfer systems that have been tailored to restricted domains and often to applications with controlled language input. The development of linguistic knowledge (grammars, lexicons, and ontologies) for such systems is an extremely time-consuming and labor-intensive process. For *unrestricted* domains, data-driven approaches can learn to perform reasonably well and much more quickly if extensive parallel text resources are available. In our project we need translations that are understandable, but not necessarily perfect. We also needed to develop a translation capability starting with effectively no domain-specific parallel texts and with insufficient resources to develop a knowledge–heavy system. Finally, while translation is presently limited to English and Spanish, the project aims at providing a model that is extensible for use by other agencies in other countries and other domains.

Given the above requirements and constraints, we require a translation approach that can be quickly adapted to other domains and languages. In that respect, CMU's Panlite system, which was used as the translation engine in the DIPLOMAT [8] a rapid-deployment speech-to-speech MT project, is ideal. We can bootstrap the system with out-of-domain parallel corpora, when available. The corpora are used to both provide examples of translations and to automatically extract a bilingual dictionary, if one cannot be obtained from external sources, for backing off to single-word translation. We then hand-refine the

[1] A corpus of such translation pairs, called a parallel corpus, is the essential 'training' data for CMU's EBMT system. The system does not 'learn' in the traditional machine learning sense; its training consists of processing the parallel data in such a way as to make retrieval of any part of the source sentences and corresponding part of the target sentences as fast as possible when the system is translating new input. The processing also includes determining the correspondence between fragments of parallel source and target sentences.

dictionary to add domain-specific vocabulary and translations, and incrementally improve the EBMT translator by adding in-domain training data as it is collected and giving it higher priority. In fact, as project members and associated personnel in the field use and test the system, translations attempts are logged, corrected and used to regularly enhance the system's database of translation examples. Translation improvements can also be achieved by using the system's generalization capabilities, which allow the examples to work in a broader range of situations [3,4].

3.4 Event Server and Event-Trigger-Rule (ETR) Server

3.4.1 Event Server. This server, installed at each participant's site, is responsible for accepting the event subscription and event filtering information provided by the Host site, where event definition, subscription and event filter specification take place. Although the Host site serves as a centralized site (registry) for event definition, event subscription and event filter specification, the subscription and filtering information is sent to Event Servers where the events actually occur. The information is used by the Event Server to determine which event subscribers should be notified based on the data associated with each occurrence of the event and the event filtering conditions. Thus, events are managed and processed in a parallel and distributed fashion. As mentioned earlier, notifications can be by email, cell phone message, and/or program activation. The Event Server at each site also 'listens' to events being posted by the Event Server of the other site(s) and communicates with the local ETR Server to trigger the processing of rules associated with the events. Thus, distributed rules, which enforce policies, constraints, regulations of different countries and different agencies, can be triggered for processing by the occurrences of distributed events.

3.4.2 ETR Server. This server, installed at each participant's site, is responsible for processing the rules defined at each site. A sub-component, called Knowledge Profile Manager, provides the user interfaces for defining rules and triggers and for managing the information about who subscribes to which events. Rules are specified in the format of Condition-Action-Alternative-Action. A rule would verify the condition of event data and/or some other local data to determine whether to perform the operations specified in the Action clause (if the condition is True) or the operations specified in the Alternative-Action clause (if the condition is False). Rules are defined and managed by different organizations to capture their policies, constraints, and security and privacy rules. Triggers are specifications that describe which distributed events trigger which distributed rules. Similar to events and rules, triggers are defined and managed by different collaborating organizations. Whenever the ETR Server receives an event notification from the Event Server, it identifies the triggers that link the event with rules and processes the rules, which may activate some agency or inter-agency processes, enforce organizational or inter-organizational policies, constraints, regulations, and/or security and privacy rules. In addition to the distributed management and processing of events, triggers and rules, there are other features that distinguish the ETR Server from the existing Event-Condition-Action (ECA) rule systems. They include specifications of guarded conditions, event history, and rule structures [11,14]. The Event Server and the ETR Sever have been implemented using JDK 1.4, on a Windows NT platform. The Web Server used is Tomcat 4.1.18, which support servlets. The Axis 1.0 toolkit is used to define, deploy and invoke the web services. The events, triggers, rules and event filters are defined using the GUIs provided by the Knowledge Profile Manager.

3.4.3 Example Scenarios. We shall describe a few implemented scenarios to illustrate the use of the ETR technology to enforce data privacy and security rules and to demonstrate automatic event notification, event filtering and information delivery.

<u>Security Rule in Distributed Query Processing</u>: Roles, assigned by the system to users, determine the access privileges of users who issue queries through the Distributed Query Processor. This role-based access control mechanism [13] is achieved by defining a rule with the ETR server that checks the rights of

a user with a particular role with respect to accessing the local attributes being queried. This security rule is invoked by an Access Controller sub-component of the Wrapper.

Police Arrest Scenario: We define the occurrence of an arrest by the police authority of a country as an event. This *Arrest* event is posted when a police officer enters an arrest record into a database. The record contains defendant information, witness information and victim information. A subscriber to the arrest event may provide event filtering conditions that specify the nature of the crime and the defendant's nationality that they are interested in. A privacy rule is defined to suppress the transfer of the victim and witness information to subscribers during event notification. Thus, the data violating the privacy of the victim and the witness are filtered out. Only the defendant's information is sent to all subscribers.

Person in the Watch List Scenario: When a visitor enters a country and the immigration official fills out a port of entry application form, the program posts an *Arrival* event to the ETR Server, which invokes the *WatchListCheckRule* to check if the visitor entering the country is on the global watch list of persons whose movements should be monitored. If not, the record is inserted into the local database table. If yes, the official is alerted. If the official decides to admit the visitor, the visitor's record is inserted into the database, and an event (*PEntry*) is posted to the local Event Server, which will automatically notify the event subscribers (police, military and security personnel) based on the filtering conditions they specified at the time of event subscription. The notification is sent out using the preferred notification mechanisms of the subscribers. It may automatically activate the processes of different agencies in different countries to handle the situation and coordinate their activities.

4. Summary and Conclusion

In this paper, we have motivated the need for an integrated transnational information system and presented the prototype implementation of such a system. The prototype system is the result of integrating four technologies: conversational interfaces, distributed query processing, event-trigger-rule management, and machine translation. It uses the Web services technology as a uniform framework to achieve the interoperability of its component systems. The architecture of the prototype system and the functionality and features of its component systems have been described. By using the prototype system in a number of scenarios, we have demonstrated that users in different countries, who speak different languages, can 1) use either a form-based interface or a conversation-based interface to query for data stored in different countries, 2) define and subscribe to events of interests, and receive notification by emails, short messages through cellular phones, or remote program/process activations to enact processes to coordinate the activities of different agencies, and 3) define and enforce policies, regulations, security and privacy using rules that are automatically triggered by the occurrences of events.

The implemented prototype has a number of limitations. The Distributed Query Processor needs to be extended so that changes made to the export schemas (and thus, the global schema) will not require code modifications to its components. Event notification is presently done by uni-casting; i.e., all notifications are sent out one-by-one from the site at which an event occurs. It can be very time-consuming if the number of subscribers is large. The application-level multi-casting approach should be explored. The distributed event, trigger and rule processing is very effective for enforcing different agencies' policies, constraints, etc. However, the control of who can define or subscribe to events, and who can define and modify triggers and rules should be specified and enforced too. One possible approach is to define a number of meta-rules using the same rule specification language. The investigation of these and other problems and system extensions is in progress.

Further research and development is also needed on the natural language components. The conversational interface needs to be improved to handle more complex sentences and to make better use of the context and semantics of conversations. Also, the present conversational interface is for English-speaking users. The interface for Spanish-speaking users is yet to be implemented. While many of the English-Spanish translations are understandable by native speakers, the output quality of the Machine Translation System would benefit from a larger number of translation examples. In fact, the compilation and use of a text corpus in this domain has presented some interesting challenges [5] but may create a

resource whose value extends beyond this project. Similarly, the application of the Panlite system to this domain has highlighted desirable types of processing that are missing in the current system and is spurring further research and development aimed at improving its performance in domains with limited and linguistically varied data and where new languages will need to be regularly added [6].

Acknowledgements: Research reported in this paper is funded in part by NSF awards EIA-0107686 and EIA-0131886. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielson, H., Thatte, S., and Winer, D., *Simple Object Access Protocol (SOAP) 1.1*, May 2000. Available from URL: http://www.w3.org/TR/SOAP/ . Accessed on: 04/2003.

[2] Brown, R., Hutchinson, R., Bennett, P., Carbonell, J.G., and Jansen,P., "Reducing Boundary Friction Using Translation-Fragment Overlap," in *Proceedings of MT Summit IX,* New Orleans*,* 2003.

[3] Brown, R. D., "Adding Linguistic Knowledge to a Lexical Example-Based Translation System," in *Proceedings of the Eighth International Conference on Theoretical and Methodological Issues in Machine Translation* (TMI-99), Chester, UK, August 1999, pp 22-32.

[4] Brown, R. D., "Automated Generalization of Translation Examples," in *Proceedings of the Eighteenth International Conference on Computational Linguistics* (COLING-2000), Saarbrücken, Germany, August 2000, pp 125-131.

[5] Cavalli-Sforza, V., Carbonell, J.G., and Jansen P.J.J., "Developing Language Resources for Transnational Digital Government Systems," the *Fourth International Conference on Language Resources and Evaluation* (LREC 2004), Lisbon, Portugal, May 24-30, 2004.

[6] Cavalli-Sforza, V., Brown, R.D., Carbonell, J.G., Jansen, P.J., and Kim, J.D., "Challenges in Using an Example-Based MT System for A Transnational Digital Government Project," the *Ninth Workshop of the European Association for Machine Translation* (EAMT-04), Univ. of Malta, April 26-27, 2004.

[7] Frederking, R.E. and Brown, R.D., "The Pangloss-Lite Machine Translation System,". in *Expanding MT Horizons: Proceedings of the Second Conference of the Association for Machine Translation in the Americas* (AMTA-96), Montreal, Canada, pp 268-272.

[8] Frederking, R., Rudnicky, A., and Hogan, C. "Interactive Speech Translation in the DIPLOMAT Project," the *Spoken Language Translation Workshop at the 35th Meeting of the Association for Computational Linguistics*, ACL-97. Madrid, Spain. 1997.

[9] Gottschalk, K., *Web Services Architecture Overview – The next stage of evolution for e-business*, September 2000. Available from URL: http://www-106.ibm.com/developerworks/web/library/w-ovr. Accessed on: 04/2003

[10] Kossman, D., "The State of the Art in Distributed Query Processing," *ACM Computing Surveys*, vol. 32, no. 4, December 2000, pp. 422-469.

[11] Lee, M., Su, S. Y. W., and Lam, H., "A Web-based Knowledge Network for Supporting Emerging Internet Applications," *WWW Journal*, Vol. 4, No. 1/2, 2001, pp. 121-140.

[12] Pellom, B., Ward, W. and Pradhan, S., "The CU Communicator: An Architecture for Dialogue Systems," in *International Conference on Spoken Language Processing (ICSLP)*, Beijing, China, November, 2000.

[13] Sandhu, R., Coyne, E., Feinstein, H., and Youman, C., "Role Based Access Control Models," *IEEE Computer, vol. 29, no. 2,* February 1996, pp. 38-47.

[14] Su, S. Y. W. and Lam, H., "IKnet: Scalable Infrastructure for Achieving Internet-based Knowledge Network," in *Proceedings of the International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet,* l'Aquila, Rome, Italy, July 31-August 6, 2000.

[15] Su, S. Y. W. et al., "A Prototype System for Transnational Information Sharing and Process Coordination: System Demo," the *National Conference on Digital Government Research*, 2004.

[16] Ward, W., "The Phoenix System: Understanding Spontaneous Speech," *IEEE ICASSP*, April 1991.