Managing Government Databases

Indiana's digital government project is using Web-based techniques to organize and retrieve data to help disadvantaged citizens receive benefits, job training, and placement.

Athman Bouguettaya
Mourad Ouzzani
Brahim Medjahed
Virginia Tech

Jerry Cameron
Family and Social Services Administration, State of Indiana

Widespread Internet use has strengthened the role of databases in government services. The increasing use of information technology has also contributed to boosting the US economy. Two recent reports underline the strategic investment in Internet-based technologies that deserve government agency funding. The first, a government-sponsored report on issues facing the nation at the dawn of the third millennium,[1] stresses the importance of efficient Web-accessible data integration and retrieval to both citizens and government agencies. The second, a report prepared by the President's Information Technology Advisory Committee,[2] presents a strong argument for investing in strategic IT areas, particularly data management.

The information revolution has led organizations worldwide to rely heavily on numerous databases to conduct their daily business. Because databases usually exist in broad, highly dynamic network-based environments, formally controlling the changes occurring in the information space—such as registering new information sources or eliciting cooperative tasks—poses a difficult challenge.

Applications such as digital libraries, healthcare, education, and finance require substantial Internet connectivity. We must empower novice and experienced users to submit complex queries over large dynamic database networks to elicit efficient solutions. This requirement calls for a sophisticated infrastructure that supports flexible tools for managing the efficient description, location, and access to Internet databases.

We propose using distributed ontologies of information repositories to meet this challenge. This metainformation represents the principal domain of the underlying information repositories. The ontologies group database collections that store similar information, and individual databases join and leave the formed ontologies at their discretion.

A common global ontology has been proposed to share information sources in large environments.[3,4] This ontology captures the structure and semantics of the information space. In general, the ontology acts as a global conceptual schema to formulate queries as though the user must rely on a single database schema. The underlying repositories' autonomy and heterogeneity make creating and maintaining a common global ontology difficult. In a large network of autonomous databases potentially spanning the globe, we propose a meaningful organization and segmentation of databases based on simple ontologies that describe coherent slices of the information space. These distributed ontologies filter interactions, accelerate information searches, and allow for data sharing in a tractable manner.

We propose investigating the design and implementation of distributed ontologies in the Web's context. By doing so, we seek to provide a Web-based infrastructure that lets users access all databases so transparently that they literally view each database as one homogeneous element of a larger database. We use government welfare and social services as a case study and proof of concept for our proposed techniques. These government agencies typically function as autonomous departments that provide services to indigent citizens.

0018-9162/01/$10.00 © 2001 IEEE

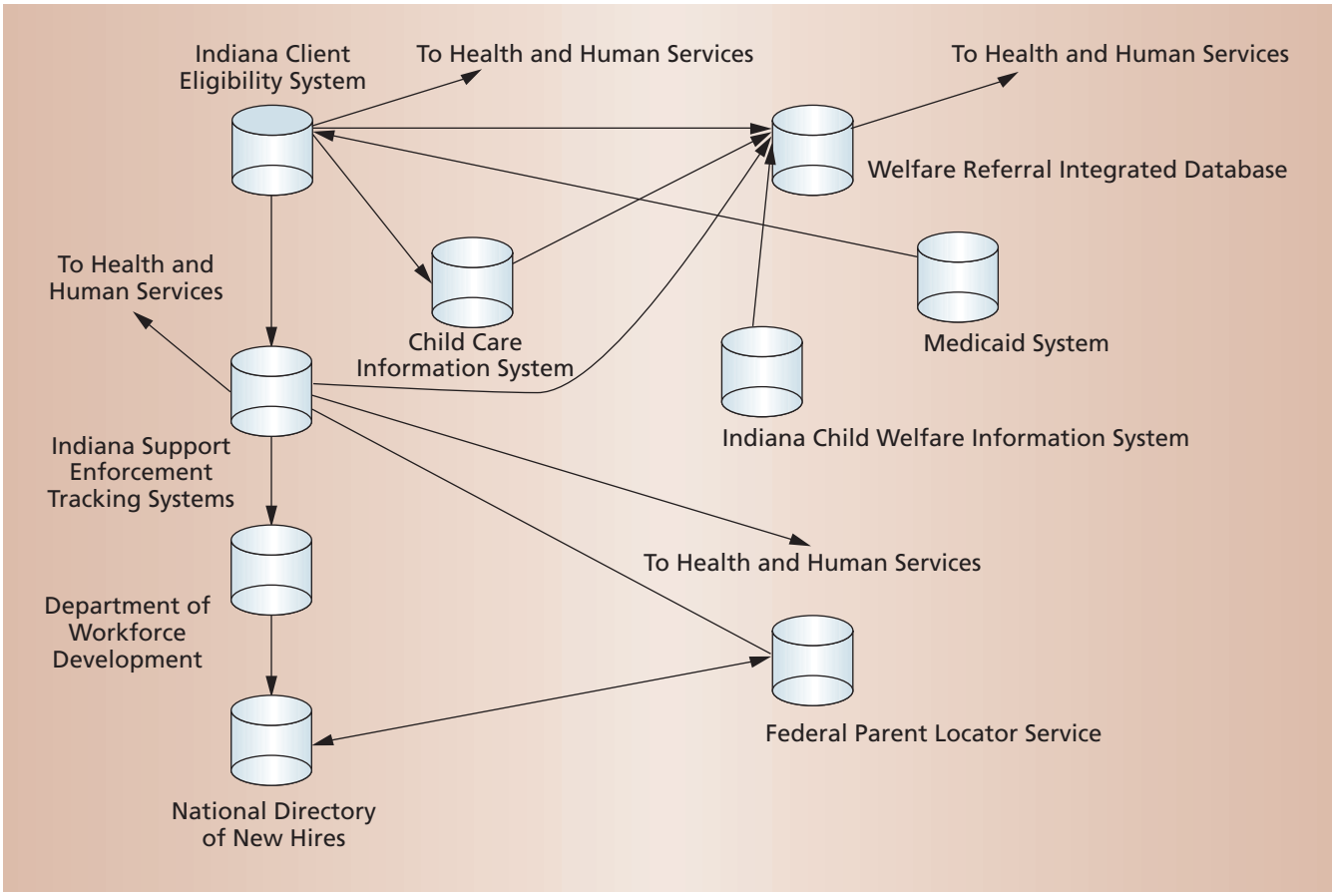

Figure 1. Database interactions to accommodate Indiana's citizens' special needs. The Family and Social Services Administration programs interact with federal and state agencies to address issues requiring interagency data access for improved planning, budgeting, reporting, and auditing.

In many cases, the current process is inefficient and costly to the agencies and citizens. Using multiple, isolated, heterogeneous, and possibly autonomous information systems that interoperate with difficulty constitutes a major problem. Thus, we have collaborated with the Indiana Family and Social Services Administration (FSSA) and the US Department of Health and Human Services (HHS) to help move their database management technology toward an effective, efficient service.

APPLICATION DOMAIN

Although our generic results reflect a range of applications, we specifically target the general area of government social services as a case study and proof of concept for our proposed techniques. In our work with the FSSA and HHS, we deal with voluminous amounts of information generated by various departments and autonomous entities.

The FSSA serves families who have issues associated with low income, mental illness, addiction, mental retardation, disability, aging, and children at risk for healthy development. The FSSA helps strengthen the families' ability to succeed in their communities. HHS protects the health of all Americans and provides essential human services, especially for those least able to help themselves. The more than 300 HHS activities include Medicare, health insurance for the elderly and disabled; Medicaid, health insurance for those with low incomes; financial assistance for low-income families; and child-abuse prevention.

We focus mainly on the FSSA's efforts to facilitate cooperation with HHS and on HHS efforts to meet federal requirements for reporting and auditing.

FSSA systems overview

The FSSA's programs target the special needs of Indiana's citizens. These state programs, interacting with their federal counterpart, address issues requiring data access from state and local government agencies. Whereas the federal agencies use this information to improve planning and budgeting, the state uses these interactions for reporting and auditing purposes. Each program usually maps to a separate information system; in turn, each system maps to several databases. In that respect, FSSA uses the primary systems shown in Figure 1.

All of these systems interact with the HHS information systems, mostly through the Welfare Referral

Table 1. Overview of Family and Social Services Administration systems providing assistance for citizens with special needs.

Agency	Application	Architecture	DBMS	Notes	Number of sites	Number of workstations
Division of Family and Children	Indiana Client Eligibility System	Mainframe	IMS	Aid to Families with Dependent Children and food stamps	140	3,800
	Indiana Child Welfare Information System	Client-server	Oracle	Child abuse tracking	95	1,200
Bureau of Child Support	Indiana Support Enforcement Tracking System	Mainframe and distributed AS/400s	DB2 and DB2/400	Child support	184	900
	Providers	Not available	Not available	Providers vary from single workstation to large server sites.	Not available	Not available
	County operations	Not available	Not available	County DFCs have many local applications using Dbase, Fox Pro, and so forth.	Not available	Not available
Division of Disability, Aging, and Rehabilitative Services	Client Rehabilitative Information System	Client-server	Not available	Vocational rehabilitation case management (planned)	30	400
	Vocational Rehabilitation Claims	Client-server	Access	Processing vocational rehabilitation claims	1	10
	Bureau of Disabilities and Determination Services	Client-server	Not available	Disability eligibility	9	100
	Bureau of Aging and In-Home Services	Islands of LANs	FoxPro	Boeing Associates developing applications for service agencies	17	100
Division of Mental Health	State operating facilities	Client-server	Not available	Financial system to support state institutions plus site administration	9	880
Bureau of Family Resources	Temporary Assistance for Needy Families	Mainframe batch	DB2	Welfare reform reporting for the Temporary Assistance for Needy Families program uses batch reporting and large file transfers.	Not available	Not available
Family and Social Services Administration	Data warehouse	Not available	Not available	All FSSA users may require data warehouse access.	400+	7,500

Integrated Database, as mandated by law. Until now, the purpose has been largely for reporting and auditing, but goals like planning and budget allocation are anticipated. Also, some systems interact with federal agencies, like the Internal Revenue Service, to intercept money owed for child support. Some state systems also interact with the Justice Department for fraud detection and legal enforcement.

Currently, interfaces between the FSSA systems and other state and federal systems differ because they lack a standard interface for data transfer and exchange. This shortcoming causes justifiable concern as it complicates the aim of developing cooperation among all these systems to maintain a low overhead. Further, multiple agencies fund these systems, which complicates the problem because they must address the issues of autonomy and heterogeneity depicted in Table 1.

Determining eligibility

Laypeople understand the concept of what constitutes a family. Unfortunately, identifying a family in the context of FSSA becomes more complex.

Using an example in which a family has a grandmother who is a single parent with two daughters helps to illustrate the difficulty of determining an individual's eligibility for benefits. In this example, the two daughters are themselves single mothers with three children each. Each child may have a different father. At least one child typically has some type of disability requiring special care.

Currently, collecting the benefits to which these disadvantaged citizens are entitled requires undergoing a time-consuming process. In many cases, dealing with this process prevents these citizens from devoting adequate time to enhancing their prospects for becoming self-supporting, for example, by seeking

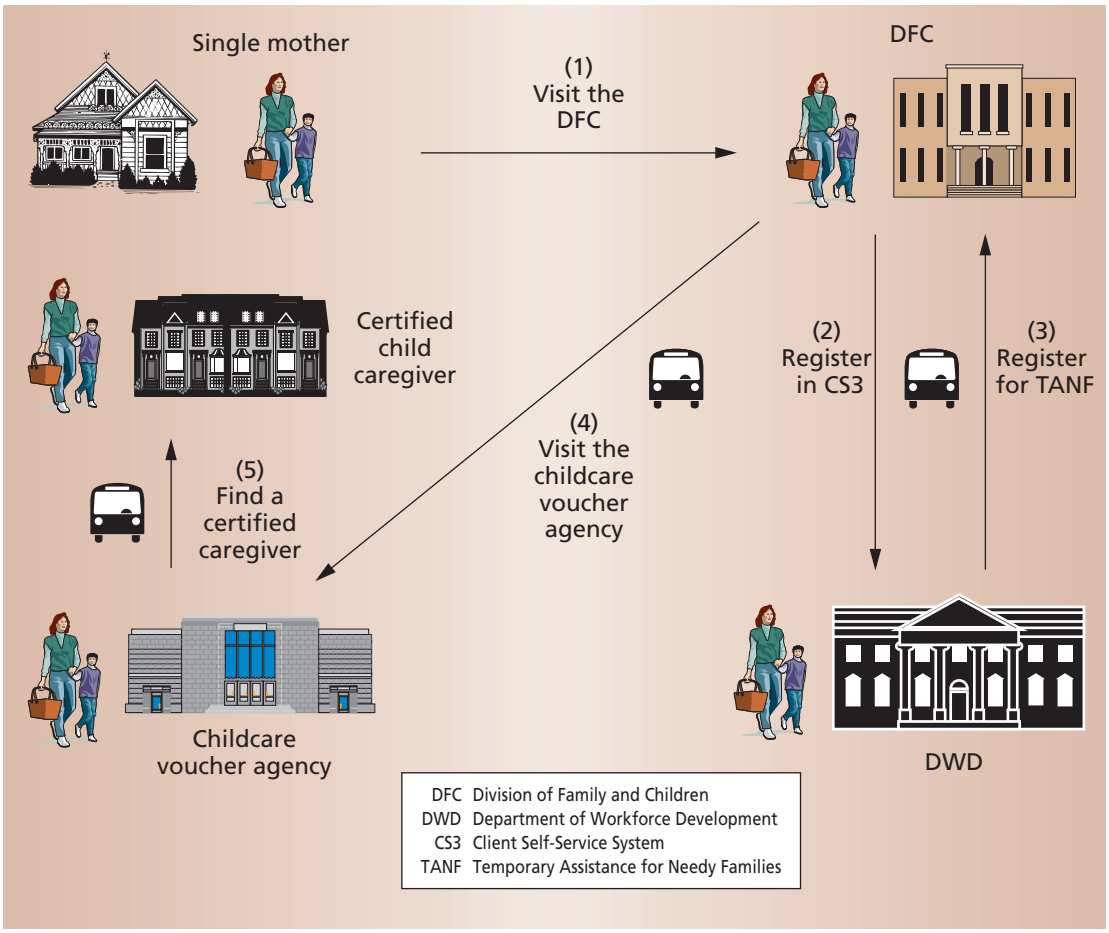

Figure 2. Benefit collection process, a time-consuming, frustrating, and complex process for needy citizens. The current system requires citizens to visit several offices in and outside the towns in which they reside to receive the benefits to which they are entitled.

professional training or pursuing an education. Essentially, the current system is so complex because the recipients must collect their benefits by visiting several offices both within and outside the towns in which they live.

In the example in Figure 2, a typical mother of a child who has a disability must first visit the FSSA's Division of Family and Children (DFC), where the case officer enters the family's personal information into the system. In our example, the mother is directed to the Department of Workforce Development to register in the Client Self-Service System. The DWD is located out of town, and the mother must travel there. At the DWD, the case officer redirects the mother to the Division of Family and Children to register for Temporary Assistance for Needy Families to receive benefits for both the disabled child and her two healthy children. The mother goes to the childcare voucher agency, which is subcontracted by the state and which is also located out of town. She then finds a certified child caregiver, who resides far from where she lives.

This process is frustrating and demeaning for both case managers and their clients. Many citizens drop out of the program, with a consequential harmful impact on the health and safety of the underprivileged adults and children who need the benefits the system is intended to provide.

MODELING DISTRIBUTED ONTOLOGIES

We adopt an ontology-based organization of diverse databases that filters interactions, accelerates information searches, and allows tractable data sharing. The criteria guiding our approach include scalability, design simplicity, and structuring mechanisms based on object orientation. Further, our approach uses an expanding network of Web-accessible databases to meaningfully organize and segment the information space using distributed domain ontologies. Information sources join and leave a given ontology based on their domain of interest. For example, information sources that share the topic "Low Income" link to the same ontology, as Figure 3 shows.

This topic-based ontology provides the terminology for formulating queries involving a specific interest, thus reducing the overhead of locating and querying information in large database networks.

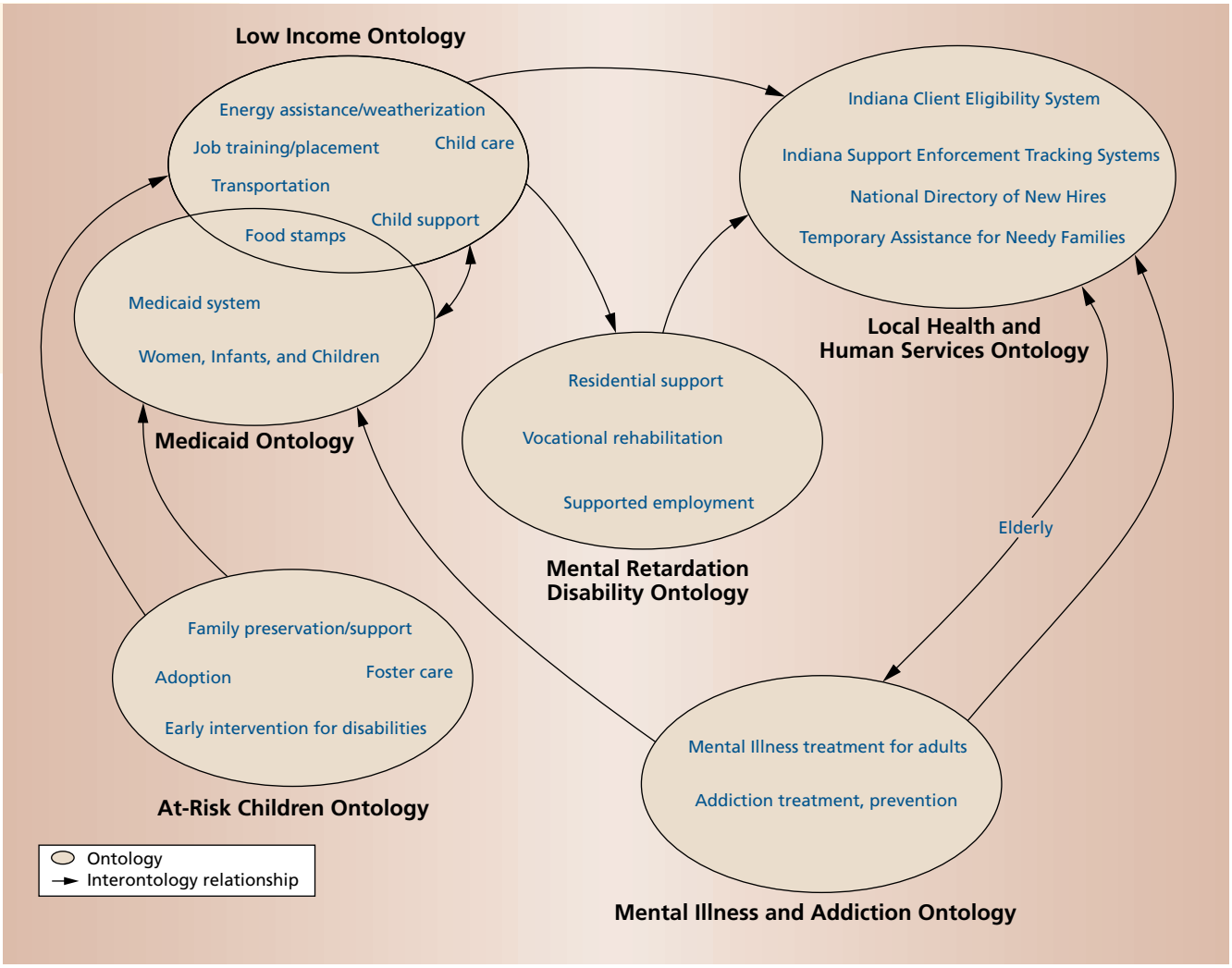

Figure 3. Example of ontologies and interontology relationships. A topic-based ontology provides the terminology for formulating queries involving a specific interest. Database is shown in blue type.

Because an information source can contain information related to more than one domain of interest, it can be linked simultaneously to more than one ontology. Interontology relationships can associate different ontologies with one another. We establish these relationships dynamically, based on user needs, and allow information sources in remote ontologies to help resolve user queries that cannot be resolved locally.

Each ontology focuses on a single common interest. It provides domain-specific information and terms for interacting within the ontology and its underlying databases. Based on common areas of interest, ontologies dynamically lump databases into a single collection, generating a conceptual space with a specific content and scope. The formation, dissolution, and modification of an ontology is a semiautomatic process. Privileged users such as database administrators have the tools to maintain the different ontologies on a negotiation basis.

FSSA/HHS application example

The FSSA/HHS application illustrates our approach's viability and demonstrates how to query the global information system. This application supports queries about related services and enables communications between many heterogeneous, autonomous social services providers and other services.

We identified nine ontologies in the FSSA/HHS application, with each ontology defining a single information type as either a service or goal. The nine ontologies are Low Income, At Risk Children, Mental Illness and Addiction, Mental Retardation Disability, Local Health and Human Services, Medicaid, Government Agencies, Law Enforcement, and Finance. For the sake of clarity, Figure 3 shows only the first six ontologies. In this example, the Elderly database does not belong to any ontology. Where two ontologies overlap, an information source stores data relevant to both ontologies. The ontology administrator initially determines the interontology relation-

ships statically, depicting a functional relationship that would change dynamically over time.

Our proposed architecture supports dynamic changes in interontology relationships. Our definition of an ontology differs slightly from definitions found in related linguistics or artificial intelligence literature. We define a concept locally, although its definition may change over time. The information systems participating in the ontology interpret what a concept means.

Co-databases

We need detailed information about each database's content in the system to locate a set of databases that fits user queries. Our approach introduces the concept of *co-databases*—metadata repositories that surround each participating database. To avoid the problem of centralized administration of information, we distribute these metadata repositories over information networks.

A co-database is an object-oriented database that stores information about its associated database, ontologies, and interontology relationships. A class in the co-database schema represents a set of databases exporting a certain type of information. A class or class hierarchy—an information type based on a classification hierarchy—also represents an ontology.

The co-database schema contains subschemas that represent ontologies and interontology relationships. The first subschema represents ontologies. The class `Ontologies Root` forms the root of the ontology tree. Every subclass of the `Ontologies` class represents the root of an ontology tree. This hierarchical organization lets us structure ontologies according to a specialization relationship.

Splitting an ontology into smaller units increases the efficiency when searching information types. The class `Ontology Root` contains generic attributes inherited by all classes in the ontology tree. The attribute `Information-type` represents the name of the `information-type`, `Low Income`, for example, for all instances of the class `Low Income`. The attribute `Synonyms` describes each information-type's set of alternative descriptions.

Each subclass of the `Ontology Root` class includes specific attributes that describe the domain model of the related set of underlying databases. These attributes do not necessarily correspond directly to the objects described in any particular database, as this example of an attributes subset for the class `Low Income` shows:

```
Class Low Income Isa Ontology Root{
  attribute string County;
  attribute Person Citizens;
  attribute set(Provider) Providers;
}
```

The second subschema consists, on the one hand, of a subschema of interontology relationships that involve the ontology to which the database belongs and, on the other hand, a subschema of interontology relationships that involve the database itself. These subschemas correspond, in turn, to two subclasses that describe interontology relationships with databases and with other ontologies.

Splitting an ontology into smaller units increases the efficiency when searching information types.

The class `Interontology Root` contains generic attributes relevant to all types of interontology relationships. For example, the attribute `Description` contains the information type provided by the interontology relationship. Interontology relationships can answer queries that the local ontology cannot.

Let's assume, for example, that the user queries the ontology `Low Income` about `Mental Retardation Benefits`. Using synonyms and generalization-specialization relationships fails to answer the query. However, the ontology `Low Income` has an interontology relationship with the ontology `Mental Retardation Disability` in which the value of the attribute `Description` is `{"Mental Retardation"}`. Clearly this interontology relationship answers the user query.

ARCHITECTURAL SUPPORT

We have explored three major directions to provide a seamless infrastructure for case managers and citizens: query infrastructure, agent middleware, and exploiting dynamic interontology relationships.

Query infrastructure

The Web's open and fluid nature forces us to specify queries generically, independent of the information source's structure and location. We provide tools to describe, advertise, and update information sources and metainformation repositories that store relevant information on available ontologies. We also provide the means to educate users about the available information space, locate the target information sources most likely to hold the information type required, and connect information sources to perform remote queries. We developed various techniques, including the novel use of documentation[5] (also referred to as demonstration) to support a specialized query language capable of seamlessly querying the data, metadata, and metametadata layers.

Agent middleware

Agents act as proxies on behalf of users to execute tasks independently and dynamically. The marriage of the Internet and databases generates complexity that amplifies the sophisticated level of potential user queries. Thus, we require an engine infrastructure that

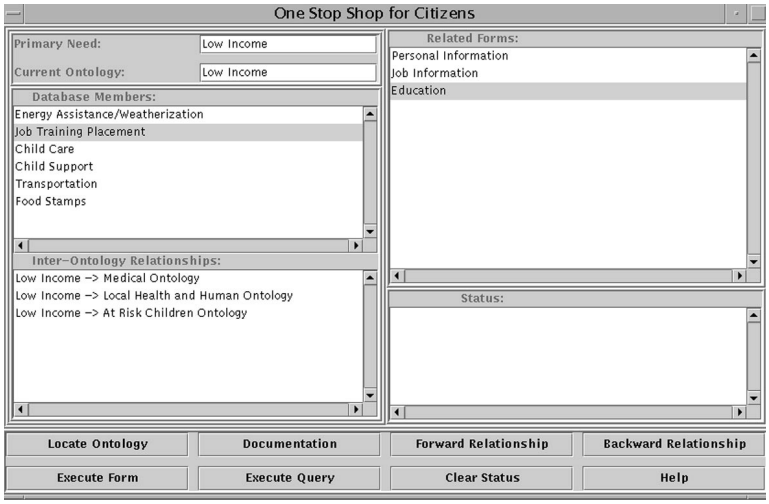

Figure 4. The case manager interface searches for relevant ontologies for the citizen's primary need, depending on the ontology's information type or synonym.

can adapt itself efficiently to changes and requirements in Web-accessible databases. Because our typical users—case managers and lay citizens—need extensive help from the developed system to locate and use the requested information, we developed an agent infrastructure to form the system engine.

We identified three categories of agents that interact with one another: user agents, system agents, and exporter agents. Co-databases serve as a relay between user queries and system agents. User agents collect information about user query history to suggest future strategies. System agents record the overall network's architecture and use statistics regarding the types of information queried and the relationships among information or parts thereof. They then communicate with exporter agents to get a sample demonstration and the actual requested information.

Registering an agent repository and replicating it across the proposed system network achieves two goals:

- The proposed system remains independent of the underlying services provided, which makes it highly extensible.
- Pushing the registry and maintenance onto the distributed computing level guarantees portability and simplicity.

Dynamic interontology relationships

Because we deal with a potentially large networked database system, a static approach to managing and organizing the information space would not scale up. Ontologies are by design inherently dynamic. Supporting dynamic relationship changes lets the system automatically notify ontology administrators when noteworthy changes in users' access patterns occur. The administrators can then decide to modify existing relationships or create new ones. They can also automatically update the related co-databases.

To implement such relationships, we use parameters to determine the relationships' strength, including using statistical data to determine how often queries traverse a certain relationship path, the type of information requested, and the distance and similarity between the source and destination. We use a formula called *heat* to determine a relationship's overall strength by weighing each parameter. We use co-databases to calculate the heat automatically, with little human intervention. During its lifetime, the heat formula changes depending on newly accumulated knowledge.

APPLICATION SCENARIO

Citizens who visit the FSSA have specific needs: They may be unemployed, unable to support their families, or need to care for disabled children. Using the ontological approach helps case managers notify relevant FSSA services so that the applicants receive all benefits to which they are entitled.

The case manager searches for relevant ontologies based on the citizen's expressed primary need. Based on the information implied by the different interontology relationships, the system connects to either the local ontology or a remote ontology. To determine relevance, the system matches the citizen's primary need with the ontology's information type or one of its synonyms.

In the example in Figure 4, the primary focus, Low Income, corresponds with the local ontology. The screen displays all databases and interontology relationships related to the located ontology. To become familiar with a particular database's content or behavior, the case manager requests its documentation.

The system also provides a list of forms for each database. We use each form to gather information about the citizen in the current database's context. For example, selecting the Job Training Placement database in the Low Income ontology reveals that up to three forms may be needed, as Figure 4 shows. The case manager could, alternatively, submit queries to a particular database in its native query language.

After filling out all required forms, the case manager can traverse the different interontology relationships to find relevant ontologies. This approach offers a flexible mechanism for browsing through and discovering potentially relevant databases. The case manager then submits requests for all benefits to which a citizen is entitled. Only those forms relevant to the citizen's situation must be completed.

Citizens can use this system to inquire about the status of their requests. This feature requires that each database provide two functions, Status and Directives, accessible through the system. The database administrator decides whether to provide these two functions and what results to return.

Enabling technologies

The implementation relies heavily on the Common Object Request Broker Architecture (Corba), Distributed Component Object Model, remote method invocation (RMI), and Java technologies. For early results, we initially focused on Corba technology,

Related Work

Several researchers are pursuing work in enabling technologies for digital governments that parallels our efforts.[1-3]

Digital Government Projects

A recent proposal[4] suggested that citizens should be able to easily access federal statistical information. The proposed approach defines a multilayered architecture based on a metadata intermediary layer that links the user interface with underlying databases. A finder module lets users specify data needs and retrieve appropriate tables and accompanying metadata.

Another proposal provides for a system that maintains and administers welfare laws,[5] using dissemination and administration modules. The dissemination module provides guidance to caseworkers and citizens to determine the regulations' applicability. The administration module gathers data for applying rules and regulations to determine their eligibility.

The regulation broker project[5] provides regulatory information dissemination. It assists companies in tracking, accessing, and using regulations over the Web. A regulation broker lets users enter information about their activities and receive the broker's replies regarding relevant regulations. Regulatory agencies provide electronic regulations' forms to the broker. These agencies maintain ownership of the documents and update their regulations as needed.

The National Institute of Statistical Sciences' digital government project[6] proposes a system that facilitates access to federal data sources and preserves data confidentiality and citizens' privacy. The system answers queries by performing statistical analyses on data extracted from federal databases. The queries are history dependent in that each query's response depends on the history of previous queries and replies.

Data Sharing Using Heterogeneous Information Sources

Integrating agent technology, domain ontologies, and information brokering, the InfoSleuth project[7] presents an approach to retrieve information and process data in a Web-based environment. Although this system's architecture deals with scalable information networks, the system does not provide facilities for user education and information space organization.

Information Manifold[8] provides uniform access to the Web's heterogeneous information sources. This domain-model component describes the browsable information space, including a domain's vocabulary, the information source's contents, and querying capability. The domain model constitutes the global ontology shared by users and information sources. Such ontologies are difficult to create and maintain because of the underlying Web repositories' variety and characteristics.

SCOPES[9] (Semantic Coordinator Over Parallel Exploration Space) is a semantic reconciliation system that uses a nonmonotonic reasoning process. This system uses ontologies to identify appropriate resources and obtain similarity mappings.

References

1. A. Bouguettaya, B. Benatallah, and A. Elmagarmid, *Interconnecting Heterogeneous Information Systems*, Kluwer Academic, Boston, 1998.
2. A. Bouguettaya, ed., *Ontologies and Databases*, Kluwer Academic, Boston, 1999.
3. dg.o, "Grant Recipients," http://www.diggov.org/about/GrantRecipients/index.cfm.
4. G. Marchionini, "Citizen Access to Government Statistical Data," http://istweb.syr.edu/~tables.
5. E. Subrahmanian and J.H. Garrett, "Two Experiences in Digital Government," http://www.ctg.albany.edu/research/workshop/background.html.
6. A.F. Karr, "A Web-Based Query System for Disclosure-Limited Statistical Analysis of Confidential Data," http://www.niss.org/dg.
7. R. Bayardo et al., "InfoSleuth: Semantic Integration of Information in Open and Dynamic Environments," *Proc. ACM SIGMOD Conf.*, ACM Press, New York, May 1997, pp. 195-200.
8. A. Levy, A. Rajaraman, and J. Ordille, "Querying Heterogeneous Information Sources Using Source Descriptions," *Proc. VLDB Conf.*, Morgan Kaufmann, San Francisco, Sept. 1996, pp. 251-262.
9. A. Ouksel and I. Ahmed, "Ontologies Are Not the Panacea in Data Integration: A Flexible Coordinator to Mediate Context Construction," *Distributed and Parallel Databases*, Jan. 1999, pp. 7-36.

which provides a robust object infrastructure for implementing distributed applications. We constructed these applications seamlessly from their components; we hosted these applications on different network locations and developed them using different programming languages and operating systems.

We download Java applets onto the user machine to communicate with system components, for example, Corba objects. In addition, with regard to distributed applications, several Java technologies such as Java RMI, JavaBeans, and Java Database Connectivity are relevant because they provide object and database access services.

We learned several lessons from this project, the most important of which is the humbling experience of becoming aware of the acute social problems that face disadvantaged citizens and those responsible for helping them. Undoubtedly, advances in computing in general and Web-based data management in particular—as exemplified by the booming Web economy—could alleviate important socioeconomic problems and help citizens preserve their dignity in time of need so that they can become self-reliant through training and job placement.

We adopted the FSSA privacy policies in our project without attempting to homogenize them across agencies. One focus of our future work seeks to develop heterogeneous privacy enforcement techniques for digital government applications.

We anticipate that privacy issues will take center stage once the enabling technologies are in place. The biggest challenge is designing a consistent yet heterogeneous set of implementable policies that will fully guarantee citizens' privacy in the face of often chaotic and inconsistent local, state, and federal privacy regulations. ✷

Acknowledgment

This work was supported, in part, by grant 9983249-EIA from the National Science Foundation's Digital Government program.

References

1. "Information Technology for the Twenty-First Century: A Bold Investment in America's Future," working draft, Jan. 1999, http://www.ccic.gov.
2. B. Joy and K. Kennedy, "Information Technology Research: Investing in our Future," *President's Information Technology Advisory Committee Report to the President*, Feb. 1999, http://www.ccic.gov/ac/report.
3. R. King and R. Hull, "I3: Intelligent Integration of Information—Reference Architecture for I3," University of Colorado at Boulder, 1994.
4. G. Wiederhold, "Intelligent Integration of Information—Foreword," *J. Intelligent Information Systems*, June 1996, pp. 93-98.
5. A. Bouguettaya et al., "WebFindIt: An Architecture and System for Querying Web Databases," *IEEE Internet Computing*, July/Aug., 1999, pp. 30-41.

Athman Bouguettaya *is an associate professor and program director in the Department of Computer Science at Virginia Tech. His research interests include Web/Internet databases, digital government, wireless computing, and electronic commerce. He received a PhD in computer science from the University of Colorado at Boulder. Contact him at athman@cs.vt.edu.*

Mourad Ouzzani *is a PhD candidate in the Department of Computer Science at Virginia Tech. His research interests include query optimization for Web databases, digital government, and e-services. He received an MSc in computer science from the University of Algiers. Contact him at mourad@vt.edu.*

Brahim Medjahed *is a PhD candidate in the Department of Computer Science at Virginia Tech. His research interests include agent architectures in electronic commerce. He received an MSc in computer science from the University of Algiers. Contact him at brahim@vt.edu.*

Jerry Cameron *is the director of the Indiana Office of Architecture and Standards, Division of Organizational Development in the Family and Social Services Administration. His interests include the use of IT techniques for digital government applications. He received a BSc in history from Ball State University. Contact him at jcameron@fssa.state.in.us.*