TELEMEDICINE AND e-HEALTH
Volume 11, Number 3, 2005

Semantic Interoperability in Telemedicine through Ontology-Driven Services

PROBNAB GANGULY, M.A. Biomed. Eng.,[1] PRADEEP RAY, Ph.D.,[2] and N. PARAMESWARAN, Ph.D.[1]

INTRODUCTION

Telemedicine involves the integration of information, human–machine, and healthcare technologies. As different modalities of patient care require applications running in heterogenous computing environments, software interoperability is a major issue in telemedicine. Software interoperability may be defined as"the capability with which two or more programs can share and process information regardless of their implementation language and platform."[1] The development methodology for interoperable telemedicine systems has been outlined.[2] Ontology can be defined as a set of concepts understood in a knowledge base. A formal ontology specifies a way of constructing a knowledge base about some part of the world and, thus, contains a set of allowed concepts and rules that define the allowable relationships between concepts. In other words, it is a structured repository for knowledge, consisting of a collection of knowledge elements such as rules and their associated data model. Ontology has emerged as a major technique for software interoperation. In this paper we highlight the importance of ontology from the perspective of semantic interoperation, describe the issues in software interoperation, highlight the importance of ontology in software interoperation, describe the major elements of our framework, describe some related efforts,[3] and describe the scope of further research in this area.

SOFTWARE INTEROPERABILITY

Software interoperability may be defined as the ability of multiple software components to interact regardless of their programming language or hardware platform. The available levels for software interoperability can be ordered on the basis of their complexity, as follows:

- Physical interoperability. In this approach, interoperability is achieved by physically transferring the information through electronic media such as floppy disks, CDs, and magnetic tapes.
- Data-level interoperability. Distributed and disparate programs support structured exchange of information through Application Programming Interfaces (APIs) invoked over a computer network.
- Specification-level interoperability. Same as the previous, and also encapsulates difference in knowledge representation at the level of abstract data types. This would enable programs to communicate at higher levels of abstraction and increases the degree of information hiding. CORBA and Enterprise Java Beans (EJB) fall into this category.
- Semantic-level interoperability. Unlike the above two types of interoperability, which are concerned with the form (structured description) at the integration interface, semantic interoperability represents design intent and predicted behavior as well as form (structured description) of the shared enti-

[1]School of Computer Science and Engineering, and [2]School of Information Systems & Management, The University of New South Wales, Kensington, NSW 2052, Australia.

ties. In this case the meanings (semantics) of different software entities need to be resolved. For example, employee or staff may refer to the same concept. Again, the term "apple" may mean different concepts such as fruit or computer to different users. Similarly, one system classifies a person as male or female, while the same person may be classified as employed or unemployed in another system. Currently many researchers are working on semantic interoperability to achieve a transparent "semantic web."

Figure 1 shows software interoperability at the three most complex levels.

Semantic interoperability issues

Semantic interoperability has been defined in general as "the ability of a user to access, consistently and coherently, similar (though autonomously defined and managed) classes of digital objects and services distributed across heterogeneous repositories, with federating or mediating software compensating for site-by-site variations."[4]

It is very difficult to represent semantic information outside a specific domain. A specific action can have different results depending on the context. For example, in a telemedicine application, the action "follow a normal diet" will mean different things to different users in different countries.

The solution proposed here is based on the concepts of ontologies and software agents. The major elements of the proposed system are:

- Because contexts would represent the underlying semantics of the databases, these contexts must be defined in the ontology, so the users only need to express their needs by using terms in the ontology.
- There are software agents collaborating with every (human) user in the distributed environment as well as the ontology and the databases. These software agents are considered similar to human entities such as, having beliefs, desires commitments and being constrained by pragmatics, considerations, and intentions; as well as semantics (meanings, propositions, validity, etc.) and syntax (formal structure, data, etc.). They handle all communication aspects, receive feedback from the user, initiate communication, monitor events, and perform specified tasks.

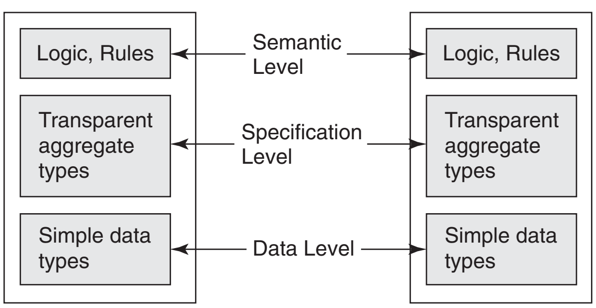

FIG. 1. Software interoperability levels.

Classification of semantic interoperability problems

We will use the following classifications[5]:

1. Semantically equivalent concepts:
   - Different terms to refer to same concept in two models, such as "staff" and "employee";
   - Different properties, such as one model includes the color of the product and the other does not; and
   - Property mismatch such as different units.

2. Semantically unrelated concepts:
   - Conflicting terms—the same term may be chosen by two information systems to denote completely different concepts. For example, the word "apple" may be used to denote either a "brand" of hardware or a type of fruit.

3. Semantically related concepts:
   - Generalization and specification—one system might have only the general concept of fruit, but the other has the concepts of mango, cherry, and so forth;
   - Definable terms—a term may be missing from one information source, but it is defined in another information source;
   - Overlapping concepts—for example, "child" in one information source means

persons aged between 5 and 12 years, but in another information source "child" may mean persons between the ages of 3 and 10 years, and in still another information source "young persons" may refer to persons aged between 10 and 30 years of age; and

- Different conceptualization—as described above, for example, one information source classifies a person as male or female. The other information source classifies the same person as employed or unemployed.

It is necessary to narrow down to an application area where one can identify above types of semantic heterogeneity. For illustrative purposes, we have chosen diabetes management, which exhibits the following semantically equivalent concepts:

- Property mismatch, such as different units for blood sugar measurement, for example, mmol/l (milimoles per liter . . . the world standard unit for measuring glucose in blood or mg/dl (milligrams per deciliter . . . the measure used in the United States).
- Different properties. Glycohemoglobin (GHb) is formed by a nonenzymatic interaction between glucose and the amino groups of the valine and lysine residues in hemoglobin. Formation of glycohemoglobin is irreversible and the level in the red blood cell depends on the blood glucose concentration. Thus, measuring glycohemoglobin provides a measurement of glycemic control over time, and its use has been proven to evoke changes in diabetes treatment, resulting in improved metabolic control. First introduced in the 1970s, it is now accepted as a unique and important index of metabolic control. There are currently four principal glycohemoglobin assay techniques (ion-exchange chromatography, electrophoresis, affinity chromatography, and immunoassay) and about 20 different methods that measure different glycated products and report different units.[6]
- Different terms such as intensive glycemic control or tight glycemic control may be used to refer to the same concept.

Similarly other types of semantic heterogeneity can be identified in this application scenario.

ONTOLOGY AND SEMANTIC INTEROPERATION

Ontology is a formal, explicit specification of a shared conceptualization that provides a vocabulary of terms and relations with which to model the domain. It is suited to represent high-level information requirements to specify the context information in a collaborative environment.

The word ontology derives from the Greek words "ontos" (which means "being") and "logos" (which means "word"). Basically it provides a formal specification of the terms in a given domain as well as a relationship among them. Thus, ontology is a formal, explicit specification of a shared conceptualization,[7] and it is related to various kinds of things that may exist in a given domain.

- "Conceptualization" refers to an abstract model of some scenarios in real life that identifies the relevant concepts of that phenomenon.
- "Explicit" means that the type of concepts used and the constraints on their use are explicitly defined.
- "Formal" refers to the fact that the ontology should be computer readable.
- "Shared" reflects the notion that ontology captures consensual knowledge, that is, it is not restricted to some individual, but accepted by a group.

An ontology should provide a vocabulary of terms and relations with which to model the domain, whereas, domain ontology captures the knowledge valid for a particular type of domain. It includes abstract concepts, and specifics domain-level constraints that can be used for reasoning, and is especially suited to represent high-level information requirements. Within schemas and classes are data level concepts that are implementation platform dependent. They are designed to optimize procedural operations. Constraints at this level are operational constraints.

A knowledge base consists of an ontology and instances of the associated classes. As the built-in knowledge in ontology is consensual in nature, it can be shared. In the practical sense, development of a domain ontology involves:

- Identification and definition of classes,
- Arrangement of the classes in a taxonomical hierarchy,
- Attaching properties (slots) and defining allowed values (facets) for these slots,
- Allocation of the values for slots for instances.

As ontologies are described in a logic-based language, the usual features of logic are available. Hence, accurate, distinct, and logical linkage, as well as distinction, can be made among the classes, properties, and relations. This would help inference engines to conduct automated reasoning. Similarly, support is provided for intelligent entities such as software agents. Ontology is a major element for representation of the semantics of the document. Thus, ontology can be effectively used as a mechanism for semantic interoperation in telemedicine as well. To illustrate this aspect of ontology, we will first describe a telemedicine application in diabetes management and then highlight importance of ontology as a means for semantic interoperation.

TELEMEDICINE AND DIABETES MANAGEMENT

Diabetes is a condition in which the level of blood glucose is consistently above the normal range. If left untreated, the condition can lead to renal and ocular complications or damage to peripheral nerves.[6] Worldwide, about 135 million people suffer from diabetes. The figure is projected to grow by 122% within the next 25 years. Hence, there is a growing need for an efficient and effective treatment plan for those who suffer from diabetes.

Currently, treatment of diabetes entails limited patient contact with multiple health-care professionals, including general practitioners and specialists. The interaction might involve regular patient visits to a general practitioner or health provider who collects blood samples to send for on-site testing. The result is reviewed by the general practitioner to determine the appropriate treatment. Complicated cases are typically referred to a specialist.

The above interaction represents one type of care currently available for diabetic patients. Two major problems identified in the current system are as follows:

- The patient, laboratory clinicians, primary care provider, or general practitoner and the specialist interact effectively among each other to provide efficient health care to the patient.
- A new approach will be established to maintain consistency within a heterogeneous environment.

We propose an ontology-driven, multiagent system for diabetic treatment. This system aims to manage complex human interactions by the use of a diabetes ontology and agent system. The major components of this implementation are: Foundations of Intelligent Physical Agents (FIPA) Standard-based Ontology development using Protege and the Toshiba Beegent platform.

The next section presents the need for an ontology-driven agent framework for diabetic management.

ONTOLOGY-DRIVEN SOFTWARE BASED FRAMEWORK

A software agent is a program that performs a specific task on behalf of a user, independently or with little guidance. A software agent also facilitates the communication of programs written in different languages. In the diabetes scenario, each health-care professional may use different systems to assist in the management of patient records and treatment plans, and each system may reside in different platforms across a network. To unify these systems effectively, a software agent could be used. The agent would create a "wrapper" around each application and communicate with other wrappers using an agent communication language. This agent-based system would help health-care professionals benefit by supporting existing health-care frameworks.

The behavioral constraints on software agents can be defined so that the agent can change behavior based on other dynamic components within the system. For example, agent technology can be used to enhance general practitioner (GP)–specialist collaboration. A typical GP–specialist interaction would involve:

- GP contacting specialist about patient;
- Specialist formulating treatment plan based on patient information;
- GP formulating treatment plan based on patient information; and
- GP and specialist negotiating ideal treatment plan for patient.

An agent system can be used to try and emulate part of this behavior independently of the real GP or specialist. This would require not only the development of an agent, but also an ontology.

The ontology helps in sharing a common understanding of the structure of information. Agents can be used to extract and aggregate information to answer queries to other applications. For instance, in diabetes management, if the GP uses the term "drug" to indicate the medicine that a patient is taking, and the specialist uses the term "treatment" to indicate the same term, a simple multiagent framework will not be able to interpret the context of each message although it has the same meaning. Ontology can facilitate the conceptual understanding of terms, and an agent can query this ontology to perform an intermediate translation of terms for use by other agent systems.

Figure 2 illustrates a simple conceptual definition for the diabetes domain. Classes (also called concepts) are the main focus of ontologies and are used to describe concepts in the domain. A class can have subclasses that represent concepts more specific than the superclass. In Figure 2, "Diabetes" represents the main concept containing specific subconcepts such as "Treatment" and "Symptoms." The Treatment concept can be further refined to represent "Medication," "Exercise," and "Diet Control." By defining relationships and constraints for these concepts, we would be able to describe the diabetes domain formally. Once

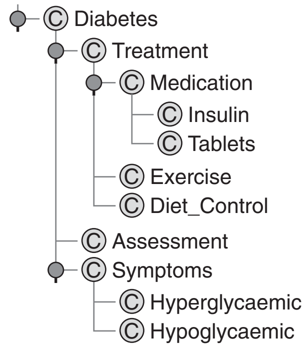

FIG. 2. Simple conceptual definition of the diabetes domain.

fully developed into a knowledge base, an agent can access these terms and relationships and to derive answers to queries. An ontology-driven, multiagent system can then be used to resolve maintaining consistency with medical terminology and standards.

ARCHITECTURE OF AN ONTOLOGY-DRIVEN AGENT SYSTEM

The conceptual model of an ontology-driven agent system is shown in Figure 3. The model is based upon the establishment of agent systems, interaction via an agent communication protocol, and development of an ontology. This model will be in compliance with the FIPA specifications explained in the next section.

The following agent systems will be established:

- Diabetes Agent System
- Ontology Agent
- Specialist Agent
- Patient Agent
- WWW Agent

Each agent performs a specific set of functions as explained below:

Diabetes agent system

This provides an interface for the GP or specialist and should incorporate an Eliza-based algorithm to respond to the user in natural language. [The technical aspects of computer programs to provide feedback have always been

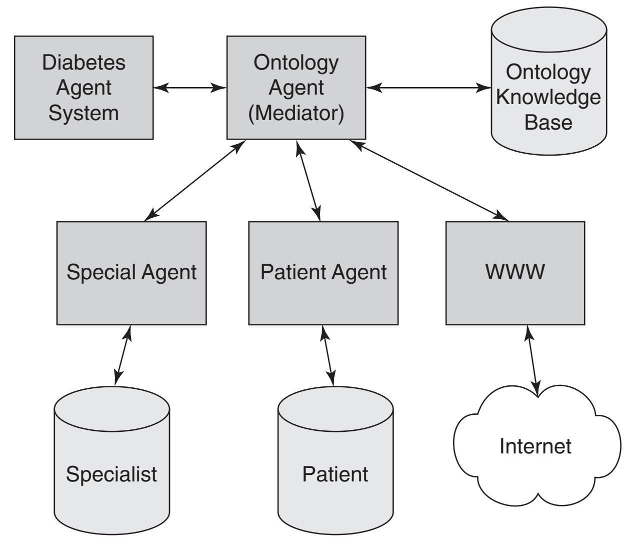

FIG. 3. Conceptual model of ontology-driven, multi-agent system.

limited. There are several programs, such as Weizenbaum's Eliza,[8] that simulate the intellect by asking a series of questions and extracting key words from the responses to build further questions or answers.]

- Sends requests from the user to the Ontology Agent.
- Receives and responds to messages from the Ontology Agent to display.

Ontology agent

- Receives messages from any agent in the system.
- Messages are broken into tokens to translate into prime concepts. This interaction takes place in collaboration with the knowledge base. For example, if a message contains a token called "web" and the primary concept defined in the ontology is "Internet," all messages containing the token web will translated into the token Internet.
- OA queries all agent systems to determine which agent can satisfy the translated message. Subsequently, the OA sends the message to the destination agent.

Specialist agent

- Provides information about treatment options for a patient. It provides access to a knowledge base to determine correct treatment options based on symptoms.
- Can negotiate with Diabetes Agent System to arrive at an optimal treatment plan.

Patient agent

- Responds to request for patient information, for example, blood glucose levels, symptoms, general patient data.
- Should interface with a patient to gather information. Could also incorporate an Eliza-based algorithm.

World Wide Web agent

- Provides access to the World Wide Web (WWW). Performs a search on data requested by other agents and launches an interface with the requested information.

System function

A scenario of how this system will function is as follows. The user (GP or specialist) requests information from the Diabetes Agent System. This request is sent to the Ontology Agent to be translated on the basis upon which the Ontology was developed. One such request could be to find information from the Internet about diabetes. The ontology agent would traverse each word in the message and translate it into a key concept. For example, the term "web" can be translated into the primary term Internet. The new message would then be: "find information from the Internet about diabetes." This would ensure that all messages passed within this system are consistent and can be understood by all agents.

After message translation, the ontology agent has to determine which agent can understand and perform the request. To do so, the ontology agent will query each agent to determine the one that satisfies the key terms in the message. If there is a match then the message is passed on to the agent.

For example, the message: "find information on the web about diabetes," would be sent to the WWW Agent. This search agent will then launch a browser to find information on diabetes.

The advantages of developing this kind of system is that we are not hard-coding "trigger words" to activate certain actions. Rather we are developing agents to deal with certain concepts within a fixed domain. The difference in concepts and trigger words is that we can define relationships and constraints for these concepts making it a more powerful alternative to current implementations. We can continually modify these concepts by simply redefining the knowledge base. If we were to hard-code trigger words, then we would be required to modify the source code of all the systems using these words. In an approach analogous to object oriented software development, we can add new and improved ontology agents without the need to modify any other system.

INTEROPERATION IN ONTOLOGY

FIPA has created a specification for the design of interoperable ontologies.[9] The purpose of developing a FIPA-compliant ontology is that it allows for the extendibility of primary ontologies. A FIPA-compliant system consists of multiple agents that interact with an Ontology Agent. The Ontology Agent responds to queries for relationships between terms or between ontologies. It could also be used for translating expressions between different ontologies. The Ontology Agent is similar to the mediator in our design. In the FIPA specifications, the Ontology Agent accessing the Knowledge Base performs the context analysis. The knowledge base can be accessed via the Open Knowledge Base Connectivity (OKBC) interface. Therefore, OKBC is used as an interface to connect the front-end Ontology Agent with the back-end Knowledge Servers.

The use of OKBC gives the added flexibility of using any existing ontology service such as Ontolingua, Loom, or Protege. OKBC also masks the back-end ontology service from the user-level agents. Using standard APIs in OKBC, the Ontology Agent can perform all the necessary ontology translations, regardless of back-end implementation.

Our design focuses on a FIPA-compliant model because of the future benefits of designing interoperable ontologies. There is also an added advantage of developing ontologies using multiple services.

IMPLEMENTATION USING BEEGENT AND PROTÉGÉ

The Toshiba Beegent[10] development framework is used for the design of the user-level agents as well as the Ontology Agent. This agent system conforms to the FIPA Agent Communication Language (ACL) specification. The ACL framework in Beegent is encoded in XML allowing for transport via HTTP.

The two main components for the Toshiba Beegent framework are the Agent Wrapper and the Mediation Agent. The Agent Wrapper is used to encapsulate an existing implementation such as the browser functionality for our WWW agent. The Mediation Agent, which is similar to the directory service, processes requests that are passed to it from an Agent Wrapper, and takes the necessary action to contact other components in the Beegent framework. The Mediator Agent will also encapsulate functionality to interface with the Knowledge Base.

DISCUSSION

Although we have used an interoperable ontology of our telemedicine application, it is unrealistic to assume a global ontology as there are varieties of standards like HL7, CEN TC 251, ISO TC 215, GEHR, etc. As more and more ontologies are developed using various standards, there will be a need to merge, align, or map between ontologies of the application domain. Unified Medical Language System (UMLS) can be considered as a large, merged ontology. More information about merging and alignment can be available from ONIONS,[11] PROMPT,[12] Chimaera,[13] FCA-Merge,[14] and SMART.[15] Mapping-related information is available in RDFT,[16] Knowledge Management Tool,[17] and MAFRA.[18]

REFERENCES

1. Howie TC, Kunz JC, Law KH. Software interoperability. Centre of Integrated Facility Engineering, Stanford University, 12/25/**1996.** (http://www.dacs.dtic.mil/techs/interop/title.shtml/.)
2. Ganguly P, Ray P. A Methodology for the development of software agent based interoperable telemed-

icine systems: a tele-electrocardiography perspective. *Telemed J* **2000;**6:283–294.

3. Ramesh V, Canfield K, Quirologico S, Silva M. An intelligent agent-based architecture for interoperability among heterogeneous medical databases. (http://hsb.baylor.edu/ramsower/ais.ac.96/papers/ramesh2.htm, accessed on 12/23/**1999**).
4. Aris OA. A framework for scalable agent architecture of cooperating heterogenous knowledge sources. In: Klush M, ed. *Intelligent information agents.* Berlin: Springer, **1999:**100–124.
5. Artemis Deliverable D3.1.1.2: Review of the State-of-the Art—Semantic Web and Web Service Semantics. The Artemis Project. http://www.srdc.metu.edu.tr/webpage/projects/artemis/, accessed March **2005.**
6. Peter JW, Paul LD, Keith WT. *Diabetes and its management*. 4th ed. Blackwell Scientific Publications, **1990**:49–53.
7. Fensel D. Ontologies: A silver bullet for knowledge management and electronic commerce. Berlin, Heidelberg: Springer-Verlag, **2001.**
8. Weizenbaum J. Eliza—A computer program for the study of natural language communication between man and machine. *Communications of the ACM* **1966**;9(1):36–45.
9. FIPA. FIPA Ontology Service Specification, 2001. (http://www.fipa.org/specs/fipa00086/XC00086D.html/.)
10. Toshiba Corp. Bee-gent. (https://www2.toshiba.co.jp/beegent/, accessed on 26/12/**2000**).
11. Gangemi A, Pisanelli DM, Steve G. An overview of the {ONIONS} Project: applying ontologies to the integration of medical terminologies. *J Data Knowledge Eng* **1999;**31:183–220.
12. Noy NF, Musen MA. "PROMPT: algorithm and tool for automated ontology merging and alignment. In: *Proc Seventeenth Natl Conf Artificial Intelligence* (AAAI). Austin, TX, **2000.**
13. Chimera (http://www.ksl.stanford.edu/software/chimera/.)
14. Stumme G, Maedche A. (FCA-Merge) Ontology Merging for Federated Ontologies on the Semantic Web. In: *Proc Intl Workshop for Foundations of Models for Information Integration* (FMII). Viterbo, Italy, September, **2001.**
15. Noy NF, Musen M. SMART: automated support for ontology merging and alignment. In: *Proc 12th Workshop on Knowledge Acquisition, Modelling and Management* (KAW). Canada, October, **1999.**
16. Omelayenko B. Integrating vocabularies: discovering and representing vocabulary maps. In: *Proc First Intl Semantic Web Conference*. Sardina, Italy, June 9–12, **2002.**
17. Crubezy M, Pincus Z, Musen MA. Mediating knowledge between application components. In: *Proc Semantic Integration Workshop of the Second International Semantic Web Conference* (ISWC). Sanibel Island, Florida, **2004.**
18. Maedche A, Motik D, Silva N, Volz R. MAFRA—A Mapping FRAmework for Distributed Ontologies. In: *Proc 13th European Conf Knowledge Engineering and Knowledge Management* (EKAW). Madrid, Spain, **2002.**

Address reprint requests to:
Pronab Ganguly
31 First Avenue
Macquarie Fields
NSW 2564
Australia

E-mail: pganguly@qantas.com.an