Using Ontology to Drive an Adaptive Learning Interface

Andrew W. CRAPO
Information and Decision Technologies, GE Global Research, One Research Circle
Niskayuna, NY 12309, USA

Amy V. ARAGONES
Information and Decision Technologies, GE Global Research, One Research Circle
Niskayuna, NY 12309, USA

Joseph E. PRICE
Information and Decision Technologies, GE Global Research, One Research Circle
Niskayuna, NY 12309, USA

and

Anil VARMA
Information and Decision Technologies, GE Global Research, One Research Circle
Niskayuna, NY 12309, USA

ABSTRACT

Intelligent, adaptive interfaces are a pre-requisite to elevating computer-based applications to the realm of collaborative decision support in complex, relatively open-ended domains such as logistics and planning. This is because the composition and effective presentation of even the most useful information must be tailored to constantly changing circumstances. Our objective is to not only achieve an adaptive human-machine interface, but to imbue the software with a significant portion of the responsibility for effectively controlling the adaptation, freeing the user from unnecessary distraction and making the human-machine relationship more collaborative in nature. The foundational concepts of interface adaptation are discussed and a specific logistics application is described as an example.

Keywords: Ontology, Adaptive Interface, Machine Learning, Autonomic Logistics, Conceptual Graphs

INTRODUCTION

For a number of years, GE has been a leader in the remote monitoring and diagnosis of complex systems such as medical imaging equipment, aircraft engines, electrical power plants, and locomotives. Increased communication bandwidth and computational power have now made it viable to integrate monitoring and diagnostics with other logistic functions such as sourcing and service planning. This integration offers the opportunity to improve overall efficiency, tuned to an organization's specific objectives, but also demands/requires that decisions depend upon an increasing volume of complex information from many sources.

It is sometimes desirable to provide collaborative decision support while keeping the human as an active participant in decision-making. In order to do this, human-computer interfaces must be more flexible, adapting to the situation and the user. This paper makes the case for adaptive interfaces, identifies some of the foundational concepts in their development, and describes an adaptive maintenance application currently under development for the Air Force Research Labs and GE Transportation Systems.

ADAPTIVE INTERFACES: WHAT AND WHY?

An interface is a boundary between two systems. The interface of concern here is that which exists between a computer-based program or agent and a human being. The majority of today's human-computer interfaces consist of a keyboard and a pointing device (e.g., mouse) for the human to make his/her intentions (input) known to the computer and a two-dimensional display screen on which the computer creates visual patterns interpretable by the human (output). Virtual reality environments can provide a significantly different interface alternative but are not widely available and are not discussed in this paper. However, the concepts developed are believed to be applicable to any human-computer interface.

Our focus is on decision support in information-rich environments such as logistics. Software applications in this arena have the potential to significantly improve the capability of the human decision maker by filtering, organizing, summarizing, and presenting what can be an overwhelming amount of data from sometimes disparate sources, and by alerting the user to critical information which might otherwise be overlooked. Summarizing, for example, can range from simple statistical analysis of numerical data to complex reasoning that logically deduces what was previously only implicit in the data and models instantiated in the computer.

The timeliness, composition, and interactivity of the information required to support synchronous decision-making varies in response to the circumstances in which action is to be taken. An adaptive interface is one that is amenable to modification in response to these changing conditions. There are many dimensions in which modification can occur, ranging from things like font size and color to language and specialized vocabulary to the level of detail and arrangement of graphical displays on the screen. What users perceive, how easily they perceive, and the interpretation they give to their perceptions depends upon the representation provided in the interface and, of course, upon the users themselves. Likewise, the ease and clarity of user input to the computer depends upon the interface provided for that input.

Information flow in either direction across the interface must make use of shared representations to encode the information. Vision pioneer David Marr[10] observed, "A representation is a formal system for making explicit certain entities or types of information, together with a specification of how the system does this." It follows that any particular representation will make explicit certain information at the expense of other information that is pushed into the background and made more difficult to retrieve. Understanding what makes information easier or more difficult to perceive is key to effective use of the human-computer interface. Adaptability of the interface is what enables communication across this interface to be made more effective for a given set of circumstances.

ONTOLOGY

Ontology hypothesizes the nature of being in a problem domain—the kinds of information that can flow across an interface. It is an inseparable part of all human rationality and communication. The world seems to us to be made up of things—call these *entities*. They may be physical, like a *rock*, or abstract, like *efficiency*. Our perceptions of and reasoning about entities lead us to conclude that some entities are more similar than others. We group similar entities together and call the grouping a *class* or a *type*. A class or type is necessarily an abstraction--it exists only in our mind as a thought (although we may communicate the thought to someone else). That which we perceive as the commonality of similar entities we call *attributes* or *properties*. We often impose a partial ordering upon our type abstractions to create a *taxonomy*. For example, *dog* is a subtype of *mammal*, which is a subtype of *animal*, which is a subtype of *living thing*. The actual, individual things that exist are called *instances*. "Lassie" is an instance of a dog.

Things in the world can interact—have an affect on each other. We call these interactions *relations* or *relationships*. "The book is on the table" identifies a spatial and possibly functional relation between the book and the table. How something participates in a relation is called a *role*. For example, Kirk is a type of *human being* (based on his intrinsic properties), but he plays the role of *captain* in relation to the Enterprise. Roles may also be organized into taxonomical hierarchies. For example, *watchers* and *listeners* are types of *sensers*, which are types of *agents* [12].

Sometimes the relations and/or attributes of one or more entities in the world, such as the book being on the table, seem to be static over some time period of interest. We call this a *state*. At other times relations and/or attributes appear to be in flux. When things are changing and we are only interested in or can only perceive the state of affairs before and after the change, we call the change an *event*. At other times we perceive a gradual change, or a progressive change through a sequence of events and states. We call this kind of change a *process*. We also use the term *role* to describe participation in an event or a process. For example, consider the statement, "The mechanic replaced the broken shroud with a new one." This describes a process with participants mechanic, broken shroud, and new shroud. Mechanic is a role, presumably played by a person.

Instances, types, attributes, relations, roles, events, states, and processes; these are the essential elements of ontology. Thus an ontology is a declaration of what does or may exist in a particular domain. It is implicit in cognition and communication, two fundamental capabilities of an intelligent system. A human language such as English is an ontology. Words and phrases represent instances, types, attributes, relations, roles, events, states, and processes. Instances, for example, may be identified by a proper noun, e.g., "George Washington," by a definite article preceding a type name, e.g., "the dog," by a definite article preceding a role name, e.g., "the president," or by an indexical such as the pronouns "he" or "it." Since natural language is often ambiguous and not computable with current technology, more constrained and therefore precise ontology languages are desirable. The role of such languages is so important that it is difficult to conceive of an artificial intelligence (AI) that does not structure its knowledge in an ontological manner so as to facilitate communication and reasoning. Reasoning by analogy, which will be discussed in more detail below, is particularly dependent upon ontological structure as the similarities in the structure itself are the basis of the reasoning. Note that an ontology may consist of multiple sub-ontologies, as will be illustrated in the application discussed below.

Semiotics is the study of how symbols have meaning [2]. Every artifact used as a semiotic sign or text, as every representation in a human-computer interface must necessarily be if communication is to occur, encompasses ontological information. The only question is whether the ontology is explicit and open to examination or implicit and assumed. The advantage of an explicit ontology is that it can be examined, communicated, used for reasoning, extended, and modified. One problem with an implicit ontology is that one has no way to know for sure what was intended or how it might be extended or modified. It is like a language without a dictionary—words have meaning and are related to other words, but there is no authority to clarify the meanings and relationships. Of course it is much less effort, at least initially, to simply assume an ontology and use it implicitly.

THE ROOTS OF ADAPTABILITY

Ontologies organize our perceptions and thoughts. This structure is the basis of reason, and reason is at the root of all intelligent adaptation. Identification of similarities between the current situation and previously encountered situations allows selection of the modifications (adaptations) that are most likely to achieve some desired result. Without the guidance of this pattern matching (reasoning), adaptations would simply be random, and would not be likely to bring about improved performance. In human-machine interfaces, the alteration can be initiated by the human or by the machine. To date, the most effective systems have generally been those that are easily adapted at the discretion of and through actions by the human users [14]. This is probably the case because humans are much better at pattern matching and opportunistic reasoning than are our present machines, and therefore much better able to decide how and when to adapt.

Let us consider some common types of reasoning. Reasoning by analogy is based on the expectation that if two structures (things, processes, situations, etc.) are similar in some respects, they are likely to be similar in others. Deductive reasoning uses structures so constrained that the truth of the premises guarantees the truth of the conclusion. For example, one form of deductive syllogism uses the partial ordering of a taxonomy to infer properties of instances from their types. Given the two premises: "All men are mortal" and "Socrates is a man," the conclusion necessarily follows: "Therefore, Socrates is mortal." Deductive systems with fixed rule sets are the least flexible way of controlling interface adaptability.

To know for sure that "all men are mortal" we would have had to examine Socrates, so strictly speaking deduction tells us nothing new. However, inductive reasoning and statistical methods can be used to find useful patterns in less than exhaustive data sets. Abductive reasoning generates hypotheses (models) based on data samples, which can then be applied deductively on the assumption of validity [12]. Reasoning with uncertainty associates a likelihood of the truth of a conclusion based on a recognized uncertainty of premise truth and/or a less than perfect correlation between premises and conclusions.

Traditionally, model-based reasoning has been identified as a distinct branch of artificial intelligence. A model is a set of entities and relationships in one domain that are intended to be representative of a set of entities and relationships in another domain [15, 16]. The goodness of a model can be measured in terms of the degree of homomorphism between the entities and relationships (structure) of the model and those of the modeled domain that are important to the objectives of the model [6]. Sowa [12] has observed that a model consists of two parts: 1) an ontology, and 2) a logic, which consists of rules of inference for deducing things of interest from the ontology and the instances of the present situation. This view of a model is shown in Figure 1. We observe that all of the types of reasoning discussed depend in the most fundamental way upon the entities and relationships that compose the ontological structure of the domain.

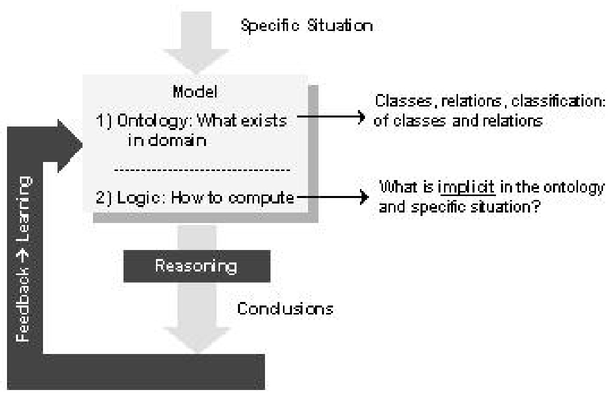

Figure 1: Two Parts of a Model with Inputs, Outputs, and Feedback

Human beings remain much more flexible and adaptive than any artifact constructed to date. One might wonder about the underlying cognitive function of the human being—what makes us so adaptable? While there are competing theories, Johnson-Laird's mental model theory has achieved considerable success in explaining human cognitive performance, both in terms of reasoning capacity and in terms of the kinds of errors committed [7, 8, 9]. According to mental model theory, each entity perceived or imagined is represented by a token in the model, the properties of entities are represented by properties of their tokens, and relations among tokens represent the relations among entities. The overall structure of the model represents a state of affairs [1, 9]. Mental model theory is compatible with model-based reasoning in that every artifactual model begins its existence as a mental model, and mental models and models captured in artifacts are both constructed of the same ontological elements. Indeed, all reasoning seems to boil down to models.

Thus we assert that models are at the root of adaptability. This truth was captured in the work of Kenneth Craik [3] who wrote, "If the organism carries a small-scale model of external reality and of its own possible actions within its head, it is able to carry out various alternatives, conclude which is the best of them, react to future situations before they arise, utilize the knowledge of past events in dealing with the present and the future, and in every way react in a fuller, safer, and more competent manner to the emergencies which face it." In the context of human-computer interfaces, we might paraphrase Craik's words by stating, "If the human and the machine share models of the problem to be solved and of their own possible actions and the probable consequences, they will be able to evaluate and collaborate about various alternatives, conclude which is the best, react to future situations before they arise, remember and utilize knowledge of past events in dealing with the present and the future, and in every way accomplish their work in a more competent manner, each performing the tasks for which it is the more adept while coordinating and integrating their efforts."

A LEARNING INTERFACE

It is argued above that effective adaptability comes through reasoning and that humans have been superior performers in controlling adaptation. This is especially the case in more open-ended problem domains such as logistics where it is not possible to pre-code models of every possible situation into the computer. Such problem domains necessitate higher-level reasoning by analogy. For a machine to be able to perform the analogous reasoning behind intelligent adaptation, it must have a store—a memory if you will—of prior situations. This is Craik's "utiliz[ation of] the knowledge of past events in dealing with the present and the future" (see previous section). Furthermore, this memory must be well organized, enabling a comparison of a current situation with prior situations and identification of those that are most similar in ways that are relevant to the current objectives. The sophistication of the memory structuring and of the pattern matching will determine the degree to which the machine is able to learn in a way useful to the adaptive process. This structuring depends on, and indeed may be a contributor to, the ontology of the problem domain.

Machine learning is a difficult problem, and our goals in this respect are modest. At this stage of our research we seek only to recognize how specific situations map on to our ontology so as to create usefully structured memories. These memories can then become the basis of reasoning as described in the application section below. However, we hypothesize that the same structuring concepts and methodologies will be found useful for higher-level learning, i.e., for extending the ontology, in future research.

REACHBACK: APPLICATION WITH AN ADAPTIVE, LEARNING INTERFACE

The application we describe implements a maintenance concept called "reachback." The term has its origins in military operations. Neal [11] defines reachback as "the electronic ability to exploit organic and non-organic resources, capabilities and expertise, which by design are not located in theater." Making better decisions through access to and assimilation of remote, asynchronous sources of information is a defining feature of a reachback capability. By applying reachback concepts to the maintenance of deployed equipment, we seek to improve the balance between the competing constraints of low maintenance cost and high equipment availability. This work is supported by Air Force Research Labs and by GE Transportation Systems. The initial prototype of the reachback system discussed here uses data from GE Transportation Systems.

GE Transportation currently monitors over 4000 locomotives worldwide. As faults and other anomalous conditions develop onboard the locomotive, they are collected and downloaded to a central monitoring center for analysis. An engineer examines this data and the output of multiple AI-based diagnostic tools to determine if the equipment is experiencing a problem. If a problem is believed to exist, a recommendation is made to have a field engineer perform troubleshooting and maintenance on the locomotive.

Although remote monitoring represents a significant increase in visibility into the equipment's condition, the right maintenance decision is not always obvious. The sensors on the equipment usually indicate "anomalies" rather than hard failures. Environmental and operating conditions have to be taken into account when distinguishing actual failures from false alarms. The importance of the mission affects whether it is practical to take the time to perform maintenance, as does the availability of personnel and parts.

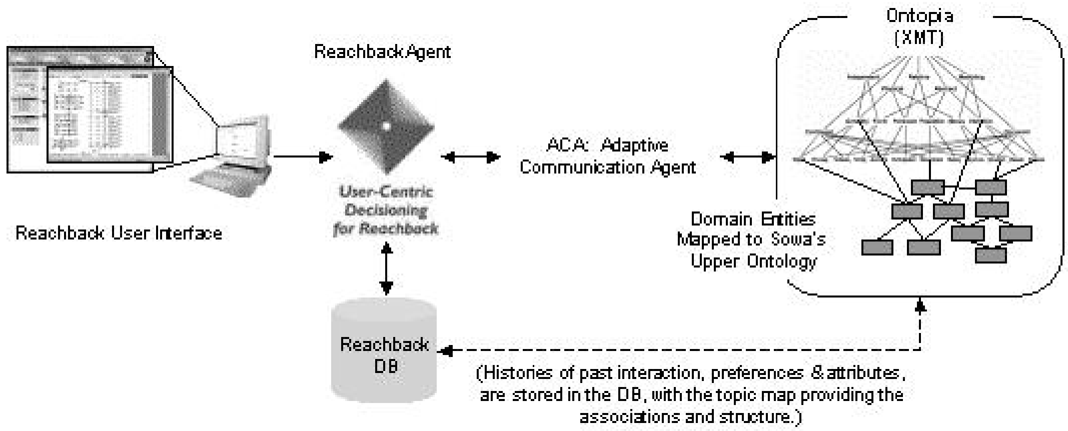

Figure 2: Architecture of Reachback Ontology-Based Prototype

Intelligent maintenance support represents a natural domain for adaptive interfaces. A reachback system must convey a large amount of sometimes-uncertain information to the decision maker, and do so in a manner that is conducive to making the best decision in the available time. While the opportunity for improved decisioning derives from this larger quantity of information, it also potentially poses the problem of overwhelming the human user with too much information to efficiently access and assimilate. The information most useful for decision support changes in response to the context in which decisions are being made. Experts in a particular subject matter, for example, are likely to have learned how to navigate and filter large amounts of raw and tool-generated data reported by a monitoring and diagnostic system. Novices, on the other hand, lack this experience and are more likely to need simpler data presentations and more supporting documentation and instructional materials. Information needs may also change according to the confidence users have in the accuracy of the data presented, which may in part be driven by their level of expertise and the confidence the system itself has in the data. Similarly, what is likely to be an appropriate level of information for a service visit in a shop may not be an appropriate level of information for service in the field, where time, tools, and experienced personnel may be more limited.

Because reachback supports knowledge-based work across a varied user population in multiple domains where new scenarios develop over time, flexible, extensible models of the human-machine interaction are needed. Several factors are important to consider in modeling this interaction.

- The nature of the information to be conveyed
- The expertise of the end user
- The preferences of the user, which may be based on sensory or cognitive capacities
- The time available to resolve a maintenance issue
- Confidence in the accuracy of the information delivered by the system

The user's expertise and confidence in decision support information are dynamic, variable and, to some degree, directly unobservable by the system. Adaptation to these contextual constraints has been identified as a key challenge in modeling the interactive process between humans and computers in a reachback system.

We hypothesize that the system is made considerably more flexible and extensible by relating concepts supporting these factors to an upper-level ontology. The upper-level ontology provides the contextual structure in which new or evolved concepts in the problem domain are placed. Such placement allows many of the attributes and associations of the new or modified concept to be inherited, greatly simplifying system evolution and extension.

Figure 2 shows the architecture of an initial reachback prototype. A "reachback agent" (RA) actively participates in determining what and how to display information in the user interface. This agent depends, in turn, on an "adaptive communication agent" (ACA) and on the data available through the reachback system. The ACA has access to both domain-specific and upper-level, domain-independent ontology represented as a topic map in XTM (XML topic map, see Garshol [5]). Some reasoning capability is provided through a prolog-like query language called tolog [4].

Figure 3 shows a screen shot of one of the reachback prototype interfaces. Fault sequences, referenced below, are displayed in the area marked with "*". Each of the other areas represents a particular type of reachback information. Several of the areas can contain information represented either as a graph, as a table, or as summary text. The type of sign used is chosen as an illustrative example of adaptability controlled by a model of human-computer interaction. Of course at a higher level, the information displayed and the layout of the information on the screen can also be made adaptive by modeling the work in which the user is engaged. Figure 4 shows the three sign types modeled as subtypes of the upper-level, domain-independent concept *schema* [12].

Human-computer interaction is a type of communication. We define communication as a process in which a participant agent or collection of agents encodes information in a sign or semiotic text and transfers these to a second participant agent or collection of agents for decoding and comprehension. Drawing on Sowa's [12] top-level ontology lattice, both *communication* and the *actions* of *sign creation*, *sign transmission*, and *sign decoding* are types of *processes*, as shown in Figure 5.

Expressed as an XTM topic map, *communication* is an *association* with three participants: an *initiator*, a *recipient*, and a *sign*. An instance is shown in Figure 6. This XTM segment indicates that this *association* is an instance of *communication*,

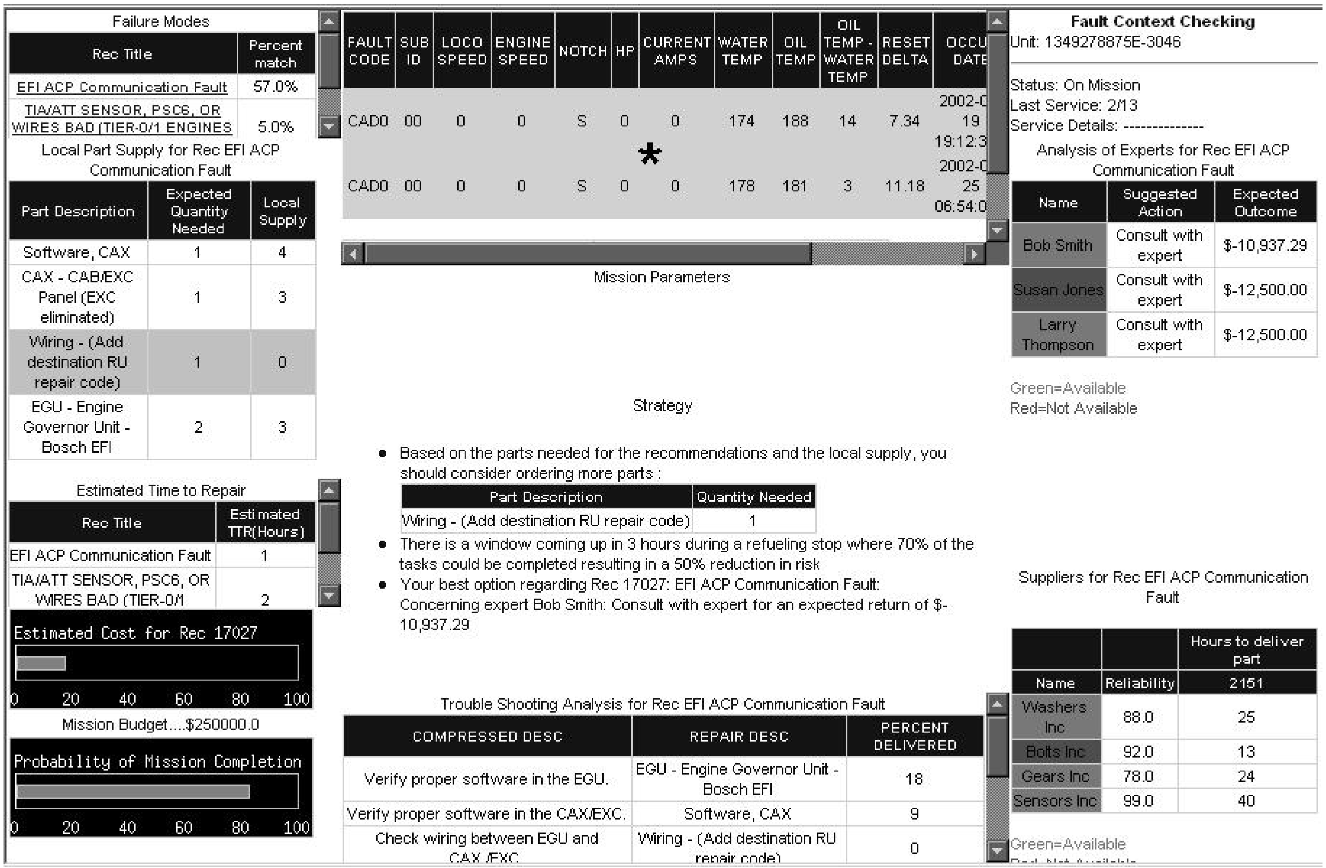

Figure 3: Reachback User Interface Screen Shot Showing Areas of Display

- All that exists
  - Abstract
    - Form
      - Schema
        - Sign Type
          - Visual Sign
            - Graphical Sign
            - Tabular Sign
            - Textual Sign

Figure 4: Visual Sign Taxonomy

has a member "georgesmith" playing the role of *recipient* (a subclass of goal) has another member "aca" playing the role of *initiator* (a subclass of agent), and a third member "graphic-93174" playing the role of *sign*. In other words, our ACA agent is the initiator of a communication with George Smith (a user) in which a particular sign was created (by the ACA).

"Reifying" the sign, meaning that it is made a topic in the topic map in its own right, captures additional information about the specific sign as shown in Figure 7. The sign is an instance of a "tabular sign" and of a "fault sequence." The fault sequence structure contains a list of faults reported by a locomotive. Faults are reported anomalies with a temporal ordering. A user may choose to view sequences of faults of varying size and beginning time. Depending upon the location and extent of this fault window, different conclusions may be drawn. In particular, the size of the window affects the observability of repeat fault codes and codes which have already been analyzed. The tabular

- All that exists
  - Independent
    - Actuality
      - Object
        - Artificial Intelligence
        - Person
      - Process
        - Action
          - Sign Creation
          - Sign Decoding
          - Sign Transmission
        - Communication

Figure 5: Communication Taxonomy

fault sequence of Figure 7 has 10 rows of data. Another association, called "relates-to" (not shown), connects the sign of Figures 6 and 7 to a particular locomotive.

The conceptual structures described above and represented in Figures 4 through 7 provide a framework in which a particular user can view a particular kind of information (e.g., fault sequences) in a particular form (e.g., as a tabular sign with 10 rows of fault codes). Simple inferencing can be used to provide reasonable default communication behaviors under a particular set of circumstances. For example, suppose user Erica Williams is examining fault sequences on locomotive SN963 for the first time, meaning that there is no historical data to indicate how she might like to see the data. The kinds of logic that might be applied include:

```
<association>
  <instanceOf>
    <topicRef xlink:href="#communication">
    </topicRef>
  </instanceOf>
  <member>
    <roleSpec>
      <topicRef xlink:href="#recipient">
      </topicRef>
    </roleSpec>
    <topicRef xlink:href="#georgesmith">
    </topicRef>
  </member>
  <member>
    <roleSpec>
      <topicRef xlink:href="#initiator">
      </topicRef>
    </roleSpec>
    <topicRef xlink:href="#aca">
    </topicRef>
  </member>
  <member>
    <roleSpec>
      <topicRef xlink:href="#sign">
      </topicRef>
    </roleSpec>
    <topicRef xlink:href="#graphic-93174">
    </topicRef>
  </member>
</association>
```

Figure 6: A Communication Association in XTM Syntax

```
<topic id=" graphic-93174">
  <instanceOf>
    <topicRef xlink:href="#tabular-sign">
    </topicRef>
  </instanceOf>
  <instanceOf>
    <topicRef xlink:href="#fault-sequence">
    </topicRef>
  </instanceOf>
  <baseName id="id307">
    <baseNameString>Loco 1 Fault List Display
for George Smith</baseNameString>
  </baseName>
  <occurrence id="id308">
    <instanceOf>
      <topicRef xlink:href="#table-rows">
      </topicRef>
    </instanceOf>
    <resourceData>10</resourceData>
  </occurrence>
</topic>
```

Figure 7: Reification of the ACA-Created Sign of Figure 4

1. Erica has recently examined fault sequences for locomotive SN385, which is the same type of locomotive. From this interaction the system learned that she prefers graphical signs. Therefore, use a graphical sign type for SN963.
2. Erica has never looked at a fault sequence before. However, she is acting in the role of a monitoring and diagnostic center engineer, and most people in this role prefer to see fault sequences as tabular signs with 8 rows of data, starting with the most recent fault. Therefore, use this display type for Erica.

Similarly, if this type of information had not been displayed before, but the ontology tells us that "fault sequence" is a subtype of "event history," we could deduce that the default communication vehicle should be the same as that most commonly preferred for other event histories. In each case we can learn over time what Erica really prefers.

OTHER ONTOLOGY APPLICATIONS IN LOGISTICS

The previous section describes part of the reachback communication model. Models of the work domain also substantially enable adaptability of the interface and effectiveness of the decision support activity. A small example of the ontology for anomaly detection is described. Conceptual graphs are chosen as a representation because of their ease of comprehension due to their compatibility with natural language and their visual characteristics. Conceptual graphs have a one-to-one mapping to first order predicate logic [13].

A conceptual graph is a bipartite graph; it consists of alternating rectangular concept nodes and oval-shaped relation nodes connected by directed arcs. A concept node must contain a type label, optionally followed by a colon and a referent label. The referent label, if present, identifies a specific instance of the type. Without referents, a conceptual graph is a generic statement of what can be. With referents, it is a precise statement of what is.

We begin by defining an observation as the measurement of a characteristic of an entity. Figure 8 shows a specific instance of

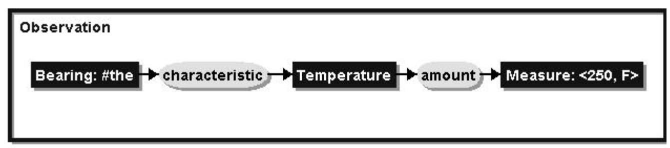

Figure 8: Conceptual Graph of a Specific Observation

an observation: "The bearing has a temperature measured to be 250 F." The bounding box identifies a context that has the type "Observation." The ordered pair "<250, F>" identifies a quantity and its units. Note the clear distinction between the temperature of the bearing and a measurement of that temperature, paving the way for data fusion, bad sensor detection, etc. Note that the existential quantifier is implied when no other quantifier is specified—the graph means, "There exists an Observation …"

Having introduced the concept of an observation, we are ready to define a symptom as a situation in which an observation is

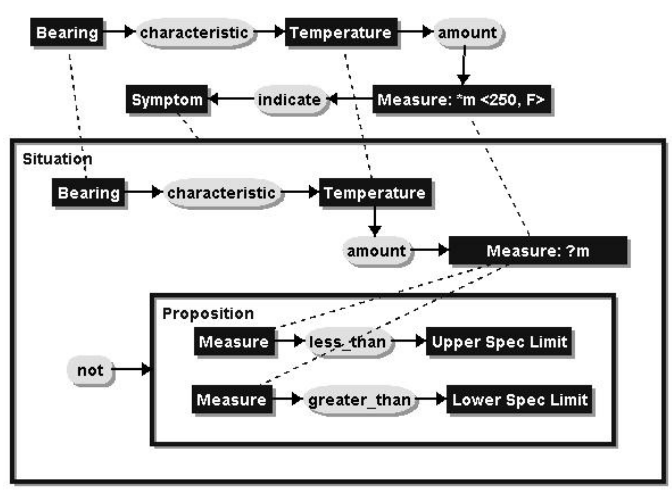

Figure 9: A Conceptual Graph of the Indication of a Symptom

abnormal. For quantitative measurements, this means that the measurement is out of spec. Figure 9 shows a more complex conceptual graph corresponding to the English sentence, “The bearing temperature measured to be 250 F is a symptom.” The dashed lines connecting nodes indicate co-referents. For example, the bearing in the context “Situation” is the same bearing that has a temperature measurement indicating a “Symptom.” The symptom, in turn, is the situation in the nested context. The innermost context is a proposition that the measure is less than the upper spec limit and greater than the lower spec limit. The “not” negates this proposition in the context of the situation that is the symptom.

CONCLUSIONS

In this paper we have discussed the value of adaptive human-machine interfaces in logistics environments and identified the kinds of ontologies, models, and reasoning necessary for a more collaborative interaction. Several aspects of an application for locomotive maintenance have been used for illustration. While much work remains to implement the kind of adaptive interface that we envision, we are encouraged that the concepts presented here constitute a promising direction for continued research.

This work was supported in part by the U.S. Air Force. We express appreciation to Jim McManus of the Air Force Research Laboratory for domain knowledge and guidance.

REFERENCES

[1] Bara, Bruno G., Monica Bucciarelli, and Vincenzo Lombardo, 2001. *Model theory of deduction: a unified computational approach*, Cognitive Science, 25, pp. 839-901.

[2] Chandler, Daniel, 1994. Semiotics for Beginners, on-line at http://www.aber.ac.uk/media/Documents/S4B

[3] Craik, Kenneth J., 1952. The Nature of Explanation, Cambridge University Press, Cambridge, UK.

[4] Garshol, Lars, 2001. *Tolog: A Topic Map Query Language*, presented at XML Europe 2001, Berlin, Germany. Available on-line at http://www.ontopia.net/topicmaps/materials/tolog.html

[5] Garshol, Lars, 2002. *What Are Topic Maps?*, Available on-line at http://www.xml.com/pub/a/2002/09/11/topicmaps.html

[6] Gurr, Corin A., 1997. *On the Isomorphism, or Lack of It, of Representations.* In Theory of Visual Languages, K. Marriot and B. Meyer editors. Springer-Verlag, pp. 288--301.

[7] Johnson-Laird, Philip N., 1983. Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness, Harvard University Press, Cambridge, MA.

[8] Johnson-Laird, Philip N., 1988. The Computer and the Mind, Harvard University Press, Cambridge, MA.

[9] Johnson-Laird, Philip N., 1993. Human and Machine Thinking, Lawrence Erlbaum Associates, Hillsdale, NJ.

[10] Marr, David, 1982. Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, W. H. Freeman and Company, San Francisco.

[11] Neal, John M., 2000. “A Look at Reachback,” Military Review, Volume LXXX, Number 5, September-October.

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

[13] Sowa, John F., 1984. Conceptual Structures: Information Processing in Mind and Machine, Addison-Wessley, Reading, MA.

[14] Vicente, Kim J., 2002. *HCI in the Global Knowledge-Based Economy: Designing to Support Worker Adaptation*, in Human-Computer Interaction in the New Millennium, John M. Carroll, ed., ACM Press, New York, NY.

[15] Willemain, Thomas R., 1994. *Insights on Modeling from a Dozen Experts*, Operations Research, 42:2, 213-222.

[16] Willemain, Thomas R., 1995. *Model Formulation: What Experts Think About and When*, Operations Research, 43:6, 916-932.