GESIA: Uncertainty-Based Reasoning for a Generic Expert System Intelligent User Interface[1]

Robert A. Harrington, Sheila Banks, and Eugene Santos Jr.
Air Force Institute of Technology
Department of Electrical and Computer Engineering
Wright-Patterson Air Force Base, OH, USA
rharring@afit.af.mil, sbanks@afit.af.mil, esantos@afit.af.mil

Abstract

Generic expert systems are reasoning systems that can be used in many application domains, thus requiring domain independence. The user interface for a generic expert system must contain an intelligence in order to maintain this domain independence and manage the complex interactions between the user and the expert system. This paper explores the uncertainty-based reasoning contained in an intelligent user interface called GESIA. GESIA's interface architecture and dynamically constructed Bayesian network are examined in detail to show how uncertainty-based reasoning enhances the capabilities of this user interface.

1. Introduction

Since generic expert systems are making their way into mainstream applications and very little research has focused on such systems, research must be conducted to handle the generic system's special challenges. The greatest of all these challenges is maintaining application domain independence, ensuring system performance is not linked to a specific domain. This challenge centers on the system's interaction with the domain, namely the system's user, so becomes the primary focus of the system's user interface.

The user interface for a generic expert system must offer more than just a pleasing, easy to use work environment. It must manage the multitude of tasks required to maintain the system's domain independence and facilitate communication between the user and the system [10] [9]. These tasks include recognizing the system's application domain, suggesting user implemented adaptations to the interface, and adapting the interface adaptively to meet the domain and user's needs [8].

In order for an interface to manage these tasks, it must contain an aspect of intelligence and reasoning capability [2], enabling it to act as a user's intelligent assistant. This is the idea behind the intelligent user interface GESIA (Generic Expert System Intelligent Assistant) for the generic expert system PESKI (Probabilities, Expert Systems, Knowledge, and Inference) [1]. This paper discusses the basic architecture and dynamically constructed Bayesian network for GESIA, focusing on the use of uncertainty-based reasoning to maintain the system's domain independence.

2. Interface Reasoning Needs

In order for an interface to perform as an intelligent assistant it must have the ability to reason. This reasoning capability is enabled by collecting metrics, transitioning metrics into information, storing information, and inferencing over the stored information. Together, these actions provide an environment from which the interface can make intelligent decisions.

The first step in creating this reasoning environment is to collect metrics, called interface domain metrics (IDMs), based on the operations being performed on the system. IDMs can be just about any type of data that the interface can collect from the domain or the user, including keystrokes, procedures used to perform tasks, and user preferences. The number and type of IDMs collected are based on knowledge required for interface reasoning. Information about the domain can be acquired from a single IDM or combinations of different types of IDMs.

The collected IDMs then need to be transformed into some meaningful information. The information format is based on interface requirements making future decisions. This step suggests an intermediate reasoning step that develops a meaning for the metric collected. This intermediate step is contained in a knowledge based transformation algorithm used by the interface to convert the metric into information.

Once the interface domain metric has been transformed, the information must be stored. The storage medium, usually

[1] This research was supported in part by AFOSR Grant #94006. A more detailed version of this paper can be found in [3].

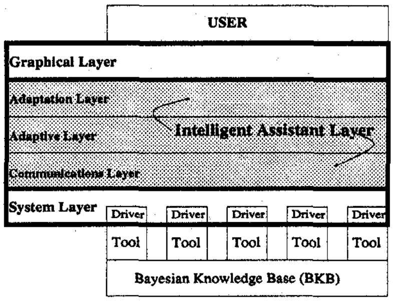

Figure 1. The GESIA architecture

a knowledge base, facilitates the reasoning process when the interface requires knowledge. An uncertainty-based scheme is a good choice for this task since it allows for efficient and effective reasoning [11] [4]. When the interface needs to make a decision, the interface will need to draw upon the knowledge stored in the knowledge base. The architectural scheme of the knowledge base will determine how intelligent and dynamic the decisions are as well as how efficient the reasoning is in terms of processing resources.

3. The Intelligent User Interface GESIA

GESIA has a layered architecture that contains three main layers: the graphical layer, the system layer, and the intelligent assistant layer (see Figure 1). The graphical layer provides the graphical interface environment, or cosmetics, for the interface, while the system layer provides a coupling between the expert system's tools and the user interface. The intelligent assistant layer is the main focus of this research.

The intelligent assistant layer has a layered architecture as well. Its layers include an adaptation layer, an adaptive layer, and communications layer. The adaptation layer recognizes particular adaptations that can be made to customize the interface to specific domains and users [6]. The recognized adaptations are then offered to the user, and the user interface provides help in making the adaptations if the user so desires. On the other hand, the adaptive layer actually makes changes to the interface without interaction or decision from the user [13]. The adaptive layer makes these changes based on perceived user behavior in a manner that will be explained later in this paper. The communications layer provides the methods to collect and translate interactions between the user and the expert system. Together, these three layers use reasoning to control adaptations, maintain domain independence, and assist the user with utilizing the system's functionality.

4. Intelligent Interface Reasoning

The basic model for representing GESIA's knowledge is a Bayesian network [7] [11] [4] [5]. User behavior is not deterministic, so representing user behavior in an uncertainty-based architecture is appropriate. This representation has the ability to portray a large amount of information based on the collection of only a small number of IDMs, making this representation important for interface reasoning efficiency. There are three types of nodes in GESIA's Bayesian network: interface learning nodes (ILNs), interface information nodes (IINs), and uncertainty support nodes (USNs).

ILNs are used by GESIA to integrate and store the meaning of collected metrics into the Bayesian network. These nodes not only hold a specific semantic meaning but also have a set of probabilistic values attached to the meaning. The actual structure of these nodes consists of a node-unique update algorithm and a table of probabilities. The node-unique update algorithm at each node is specially designed to alter node probabilities based on the type and use of the metric received. The table of probabilities holds an entry for every user of the system plus an entry for each of the four basic user types: application user, application expert, knowledge engineer, and computer scientist. The specific semantics of each ILN combined with its probabilistic values allows the ILN to represent degrees of uncertainty in the learned information.

IINs represent the many states of the world within GESIA. These nodes feed off ILNs, USNs, and other IINs to determine probabilistic values for the states they represent. While the ILNs are the primary gateway for which learning enters the network, IINs represent the application of the networks learned knowledge. Interface elicitation of knowledge targets the states represented by the ILNs, allowing the interface to make intelligent decisions concerning potential adaptations.

Finally, the USNs store information concerning the uncertainty that the user interface will make a correct decision about a particular IIN. The structure of these nodes is much like the structure of the ILNs, and there exists exactly one USN for each and every IIN. The probabilities stored in each of these nodes represents all the instances when the interface is wrong about inferencing over the IIN it supports. This uncertainty is applied to it's parent IIN to alter it's parent's probability when it's parent is targeted for knowledge elicitation by the interface. In this way, the interface decisions of the future will be affected by it's incorrect inferences of the past.

5. GESIA Metrics and Nodes

GESIA employs a *minimum* set of metrics necessary to assist the user with employing system functions while maintaining the system's domain independence. All of GESIA's IDMs, ILNs, USNs, and IINs fall within one of three general classifications based on the information collected: functional execution, communication modes, and output styles. Together, these classifications of information provide GESIA with a considerable amount of knowledge about the world.

5.1. Functional execution

The metrics and nodes used to support the functional execution classification are used to collect, learn, and use information answering the question "What functionality of the system will this user most likely use?" Every time one of the main expert system functions (knowledge acquisition, knowledge extraction, and data mining) are executed, a metric is instantiated representing that execution. This metric is sent to the appropriate functional execution classification ILN and is applied to each ILNs probability that is associated with the current user. Using each node's individual update algorithm, a new probability is produced and stored in the node's user table for the current user. Later, when the interface needs to question a functional execution IIN, the probabilities of the subordinate ILNs and IINs are applied. The information collected for the functional execution classification is used by the interface to perform interface initiated abstraction of seldom used functions and to support decisions for the communication modes classification.

5.2. Communication modes

The metrics and nodes used to support the communication modes classification are used to collect, learn, and use information to answer the question "What type of communication does this user prefer when utilizing the powers of the expert system?" A metric is collected each time a communication mode (natural language, structured text, or graphical manipulation) of the interface is used to perform an expert system task. This metric must be collected each time a communication mode is activated to translate between the user and the expert system and is processed much like the functional execution metrics. An example of the use for this information is if the user has a high probability of using a particular communication mode and the user starts an expert system function, the interface can automatically bring up that communication mode. If the probabilities are close between two modes, the system initiates a query to the user asking which mode the user wants.

5.3. Output styles

The metrics and nodes used to support the output styles classification are used to collect, learn, and use information to answer the question "How many of the best matches from a query will this user prefer?" The IDM is collected by obtaining how many outputs the user requests, or how many outputs the user accepts if a reasoned number of outputs is returned by the interface. Again, this metric is processed much like the function execution metrics. The output styles information is especially useful when the user fails to specify what style of output is required for a specific execution of an expert system query. As with the communication modes, if the interface finds probabilistic tendencies toward a particular output style, the interface will automatically return the most probable desired output style. If the probabilities are close, the user interface will query the user for clarification.

5.4. GESIA's Bayesian network

The interface domain metrics, ILNs, USNs, and IINs are combined to dynamically construct the GESIA Bayesian network [3]. This network represents the knowledge that the user interface collects dynamically, as the user utilizes the expert system. As previously mentioned, the ILNs occupy the fringe of the structure and offer a means to aquire newly learned information. These ILNs lend dependencies to corresponding

These dependencies can also be passed to other IINs. In GESIA's network functional execution IINs, ILNs, and USNs all feed dependencies into the communication mode IINs. Likewise, communication mode IINs, ILNs, and USNs feed dependencies into the output style IINs. Together, these relations add probabilities as they trace through the network to influence the final probability of the IIN being questioned.

5.5. Example of interface information node query

An example of a simple network will now be presented. This example demonstrates how the network learns and how the learned data can be used to create a probability for a possible state. Figure 2 depicts the network used in this example. Notice there is only one IIN, named "User is Using Graphical Communication" (UGC). There is also the supporting USN, named "Uncertainty User is Using Graphical Communication"(UUGC). Finally, there are two ILNs, named "User's Class Prefers Graphical Communication"(CPGC) and "User Prefers Graphical Communication"(UPGC). For this example, let's say a user, login TOM, has logged onto PESKI through GESIA. GESIA's network recovers all the learned data about TOM from storage and sends the data to the appropriate ILNs and USNs in the network.

With the network loaded, TOM begins to use GESIA. As TOM performs actions through the interface, the interface records TOM's behavior by updating network ILNs and USNs. For example, in Figure 2, if TOM chooses to use graphical communication from the communication mode menu of the interface, the interface will update CPGC and UPGC. Thus, TOM's behavior is captured.

Later, if the interface wants to guess what communication mode TOM will choose, the interface will query the UGC for the node's probability. This probability is calculated by combining the probabilities of CPGC, UPGC, and UUGC. The

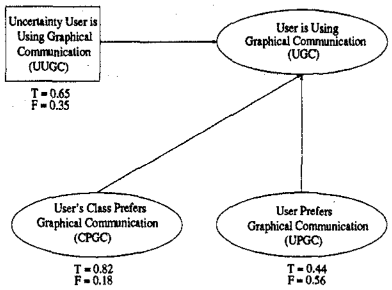
Uncertainty User is Using Graphical Communication (UUGC)
T = 0.65
F = 0.35
User is Using Graphical Communication (UGC)
User's Class Prefers Graphical Communication (CPGC)
T = 0.82
F = 0.18
User Prefers Graphical Communication (UPGC)
T = 0.44
F = 0.56

Figure 2. A partial network for query example

probabilities are combined using the following method. First, a truth table is constructed that lists all the possible combination of the truthfulness of CPGC and UPGC. Therefore,

$P(UGC=T \mid CPGC=T, UPGC=T):=1.00$,
$P(UGC=T \mid CPGC=T, UPGC=F):=0.65$,
$P(UGC=T \mid CPGC=F, UPGC=T):=0.65$, and
$P(UGC=T \mid CPGC=F, UPGC=F):=0.00$.

Notice if the probabilities that CPGC and UPGC are both true then the probability of UGC being true is 1.00, and if the probabilities that CPGC and UPGC are both false then the probability of UGC being true is 0.00. In cases when the probability of an IIN is not absolute, the uncertainty of the truthfulness of the IIN must be supported by it's USN. Once the truth table is constructed, the probabilities may be combined using Bayes theorem [7] [11] [4] [5]:

$$\begin{aligned} P(UGC=T) = {} & P(UGC,CPGC,UPGC) \\ & + P(UGC,not(CPGC),UPGC) \\ & + P(UGC,CPGC,not(UPGC)) \\ & + P(UGC,not(CPGC),not(UPGC)). \\ = {} & (1.00*0.82*0.44) + (0.65*0.18*0.44) \\ & + (0.65*0.82*0.56) + (0.00*0.18*0.56). \end{aligned}$$

Therefore, UGC-T=0.7108 or 71 percent. Given this result, the interface has acquired a mathematically sound method to capture user behavior and convert it into a representation from which the interface may reason about future user intent.

6. Results and Conclusions

Including a reasoning architecture in a user interface to facilitate intelligent interface decision making can be performed using Bayesian networks. Uncertainty-based principles enhance the ability of intelligent interfaces to adapt to application domains and users. Emphasis on generic expert system user interface design, and the reasoning architecture behind it, is important to keep generic expert systems generic.

There are still some shortfalls that must be overcome such as metric collection and information storage. The collection of too many metrics too often overburdens the tasks the user is trying to perform, reducing the value of the interface's intelligent adaptations. The collection of metrics must be controled to ensure the IINs are accessed and updated in a deterministic fashion much like the process control schemes operating systems designers utilize [12]. the user's probabilities at each of the IINs is relatively trivial when only few users employ the expert system. However, as the system gains users, more records will have to be stored at each IIN. This storage must be managed to avoid overtaking storage memory resources.

References

[1] D. O. Banks. Acquiring consistent knowledge for bayesian forests. Master's thesis, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, 1995.

[2] E. Bos, C. Huls, and W. Claassen. Edward: full integration of language and action in a multimodal user interface. *International Journal of Human-Computer Studies*, 40:473–495, 1994.

[3] R. A. Harrington, S. Banks, and E. S. Jr. The peski intelligent user interface. Technical Report AFIT/EN, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, 1996.

[4] E. S. Jr. Computing with bayesian multi-networks. Technical Report AFIT/EN/TR93-10, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, 1993.

[5] E. S. Jr and E. S. Santos. Representing and reasoning with bayesian knowledge-bases. *submitted to Journal of the ACM*, 1996.

[6] E. Kantorowitz and O. Sudarsky. The adaptable user interface. *Communications of the ACM*, 11(32), November 1989.

[7] K. B. Laskey. The bounded bayesian. In *Proceedings of the Eight Conference on Uncertainty in Artificial Intelligence*, 1992.

[8] B. Meyer. Retail user assistant: Evaluation of a user-adapted performance support system. In *Lecture Notes in Computer Science, 4th International Conference, EWHCI'94*, 1994.

[9] I. Neilson and J. Lee. Conversations with graphics: implications for the design of natural language/graphics interfaces. *International Journal of Human-Computer Studies*, 40:509–541, 1994.

[10] R. Oppermann. Adaptively supported adaptivity. *International Journal of Human-Computer Studies*, 40:455–472, 1994.

[11] J. Pearl. *Probabalistic Reasoning in Inteligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA, 1988.

[12] M. Singhal and N. G. Shivaratri. *Advanced Concepts in Operating Systems.* McGraw-Hill, Inc, New York, NY, 1994.

[13] J. E. Trumbly. Productivity gains via an adaptive user interface: an empirical analysis. *International Journal of Human-Computer Studies*, 40:63–81, 1994.