AUTOMATED EXTRACTION OF KNOWLEDGE FOR MODEL-BASED DIAGNOSTICS

Avelino J. Gonzalez Harley R. Myler
Massood Towhidnejad Frederic D. McKenzie Robin R. Kladke

Department of Computer Engineering
University of Central Florida
Orlando, FL 32816-0450

ABSTRACT

The concept of accessing CAD design databases and extracting a process model automatically is investigated as a possible source for the generation of knowledge bases for model-based reasoning systems. The resulting system, referred to as Automated Knowledge Generation (AKG), uses an object-oriented programming structure and constraint techniques as well as internal database of component descriptions to generate a frame-based structure that describes the model. The procedure has been designed to be general enough to be easily coupled to CAD systems that feature a database capable of providing label and connectivity data from the drawn system. The AKG system is capable of defining knowledge bases in formats required by various model-based reasoning tools.

1.0 INTRODUCTION

The process of knowledge acquisition has been an impeding factor in the growth of knowledge-based systems. For this reason, research in automating the process has attracted the interest of a number of investigators around the world. Although significant progress has taken place (Marcus 89), a significant difficulty has been that the knowledge required for more traditional rule-based systems is of extensive and complex domains and generally found only in the minds of human experts.

The emergence of model-based reasoning techniques in control and diagnosis of electrical, mechanical, and/or process systems has opened an avenue of opportunity in the area of automated knowledge acquisition. The knowledge required in such systems is actually a model representation of the system to be analyzed. This knowledge is not in the form of explicit rules and is extractable from schematic drawings of the target system. When such drawings exist in electronic media such as a Computer-Aided Design (CAD) system, the automation of the knowledge acquisition process simplifies.

In general, CAD databases do not provide all the information necessary to generate a complete knowledge base. Additionally, the lack of constraints placed upon the draftsperson doing the drawing requires that acquisition system be able to understand the intent of the process system model and thus make estimates of what the draftsperson intends to represent. This process is no different than that followed by a human process engineer trying to carry out the same task.

This topic is under investigation at the University of Central Florida Department of Computer Engineering. In a three year project funded by NASA-Kennedy Space Center, an objective was set to develop a system capable of generating knowledge bases from CAD databases with minimal human interaction. The prototype system is called the Automated Knowledge Generator (AKG) and is the topic of this paper.

2.0 THE AKG SYSTEM

An Object-Oriented Programming (OOP) approach using the Symbolics Genera 7 LISP machine environment has been taken in the development of AKG. Each component of the target system described in the CAD database is represented as an object within AKG. This approach is intended to model the physical system as closely as possible by representing components as an organized set of discrete objects capable of communication with external processes. In addition, OOP encourages modularity of design, thus making development, modification and enhancement of the system much simpler. The AKG system is divided into eight modules as shown in Figure 1.

The AKG process can be divided to two major tasks. 1) the capture of information which resides in the CAD database, and the creation of an internal model 2) the resolution process, which include the verification of captured knowledge and the generation of missing information.

2.1 Knowledge Acquisition from CAD

At start up, a CAD-generated description of the target system is obtained through the ACCESS module. This module communicates with the computer hosting the CAD system and downloads two files, COMPOC.DAT and TOFROMC.DAT, that must be formatted by the CAD database system. ACCESS uses a command file that contains the unique communication configurations required by the host as well as appropriate database query instructions needed to format the data files. The COMPOC.DAT file contains component details made up of a unique identifier, nomenclature, and possibly other descriptive information such as operating range and units. The TOFROMC.DAT file contains structure data which describes the process component interconnectivity in the system being modelled. The SPAWN module then uses information from the COMPOC.DAT file to create unique component objects within the AKG environment. The CONSTRAINT GENERATOR module sends connectivity information to each of these component objects. The connectivity structure imposed represents an initial constraint set on the system.

Once the CONSTRAINT GENERATOR completes its process, all the available information has been collected from CAD and an internal model is established. This internal model lacks information regarding the functionality of the source system it represents. In order to accomplish a complete knowledge acquisition, additional modules are called upon to generate the function data. Generation of the function data is termed resolution and is the primary knowledge generation process.

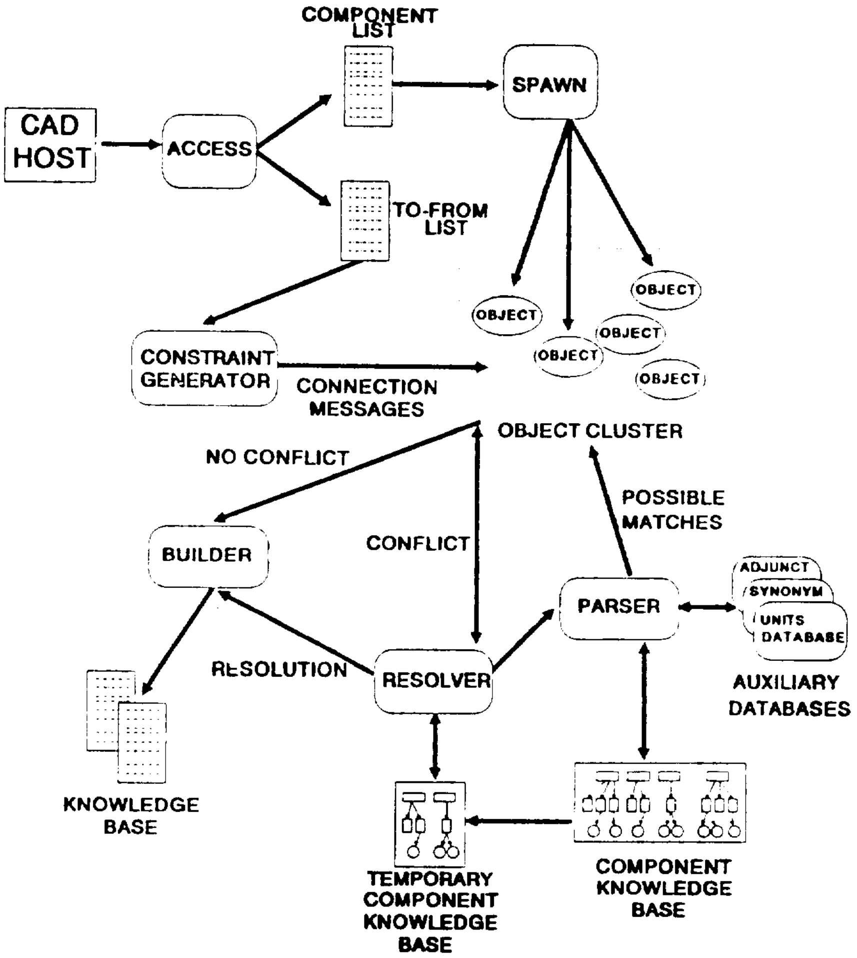

Figure 1. A graphical representation of the AKG process

2.2 The Resolution Process

In order to accomplish the resolution process, AKG uses the PARSER, COMPONENT KNOWLEDGE BASE, and RESOLVER modules.

2.2.1 PARSER

The PARSER provides the first level of identification of the components in the source system. PARSER uses several string matching heuristics (Kladke 89) to search through the COMPONENT

KNOWLEDGE BASE (CKB) in order to find one or more possible matches for each component and the label supplied from the CAD system. PARSER utilizes an internal confidence factor to rank the possible matches. A match confidence of one hundred identifies a perfect match between the source system component label and one found in the CKB. This process is a form of concept learning (Rendell 1987) because a search is made for a measure of graded class inclusion that is consistent with experience, the known CKB objects.

2.2.2 COMPONENT KNOWLEDGE BASE

The descriptive representations of components in the CAD system are not as complete as would be required for the proper operation of a diagnostic and control system. A major deficit to the completeness of some component descriptions, for example, is the lack of output functions. To complete component frames and to further resolution of the flow inconsistencies that exist in the connectivity of the CAD representation, more information is needed. An easily accessible database of generic-type components with a description of their functions and other significant data is the link to complete resolution of the source system.

The role of the COMPONENT KNOWLEDGE BASE is to provide the information necessary to complete the functional description of a component. This information includes the output function, parameters that affect the output function, and parameters that affect the performance of a component such as tolerance and delay. Descriptions of generic components that resemble a particular component in name and nature are stored in a hierarchical internal database. By determining the generic component which best fits the name and nature (analog-component, digital-component, etc.) of the specific component, the vital information known to the generic component such as output function can then be inherited by the specific or instance level component of the internal model to further enrich its own description. It is at this point that the component frame may be complete enough for use with a reasoning tool. More complete component frames also lead to better opportunities to resolve flow inconsistencies. The quantity and quality of information inherited depends on the degree of accuracy of the match. Generic components in the CKB are stored as frames and, when accessed, are spawned into internal objects. As an object, each component possesses its own identity and function.

The conceptual structure of the CKB is a list of top level generic components in the knowledge base that access more successively descriptive components. Upper level components constitute types of devices and have information that govern the accepted behavior of these device types. This information is carried through to the children of these upper level devices as a result of inheritance during the spawning process.

The process of CKB access can be broken into the following stages:

1. The path of a component (i.e., the generic component to be accessed along with its ancestors to the top level) is given as an argument to the access function.

2. The access function retrieves the generic component from storage and allows inheritance from its parents.

3. Once retrieved, the generic component is spawned into the AKG world the AKG world as a generic comp

4. A list of the children of this component is returned.

5. These children are used by the PARSER to further add to the depth of the path to be accessed.

6. Each accessed component is then noted within a global list that serves as a temporary component knowledge base for later use by RESOLVER.

An editor is provided that allows direct user modification of the CKB. This utility has many features including editing of both actual storage frames of generic components and spawned generic components.

An extension to the CKB is the implementation of a constraint representation scheme which will encompass process knowledge for generic components. AKG uses the criterion that process and control system components must have similar, and sometimes identical, properties. The idea is to interrelate components that belong to certain process system classes (such as electrical, pneumatic, flow, etc.). For example, one never connects a logic gate to a pressure valve. The CKB provides these properties as constraints of the components. This knowledge base contains general domain knowledge concerning component details and system aspects of process control. Such information will not only include standard values for tolerance, delay and transfer function for each generic component represented, but also will include constraints indicating which components may be validly connected. The availability of process knowledge allows the primary constraint propagation mechanism (Resolver) to further identify and select the best generic component and transfer function for a specific CAD component.

2.2.3 RESOLVER

The Resolver examines components in the system to establish an initial confidence factor for each. Each slot in the internal object cluster is assigned a weight (e.g., the OUTPUT-FUNCTION and NOMENCLATURE slots have the weight of 20, and the RANGE and TOLERANCE slots have the weight of 5). These weights are based on the amount of importance a particular slot has in determining the final identification/operation of the system. The initial CF for each component is computed by summing the weights of those slots that are filled directly from the CAD database. This value represents the level of information that a component has about itself with respect to all other objects in the system. A global threshold for the confidence factor is established by the user which, when exceeded, flags a component as ready for conversion into a knowledge-base frame. If a component's confidence factor does not reach the minimum threshold due to lack of information, the RESOLVER module is called to deduce the correct identification from the CKB. The confidence factors at each object are not independent. This is a significant difference from the way CF's are used in rule-based systems. No single CF, CF cluster or CF sequence can dominate the final outcome of the resolution process.

The RESOLVER calls PARSER with the list of inadequately identified components. Upon completion, PARSER adds a list to the POSSIBLE-MATCH slot of each component flavor for which a match was found. This list includes the component matched within the CKB and a parse confidence factor that reflects the certainty of the match. The RESOLVER searches the temporary CKB which is produced during the parsing process as a result of accessing the components in the CKB (see section 2.2.2), for the match with the highest parse confidence.

Once this component is found in the temporary CKB, the RESOLVER attempts to verify the match between the component in the system and the generic component from the CKB. This is accomplished by comparing the slot values (i.e., values for UNITS, RANGE, allowed/possible upstream (INPUTS) and downstream (OUTPUTS) components, etc.) for the component and the generic component. If a match is confirmed, the RESOLVER supplies the information missing from the system component with the information contained in the generic component from the CKB. The act of adding information to a component flavor causes an immediate increase in the confidence factor of that component.

If a match between the component and its best possible match can not be supported, the RESOLVER will attempt to match against the remaining components in the list of POSSIBLE-MATCHES. If still unsuccessful in finding a match, the RESOLVER attempts to match the component with the parents of the possible matches, starting again at the best match. As it was discussed in section 2.2.2. the parent of a generic component in CKB is a more general form of its children. In this case the RESOLVER relaxes the constraints on the possible match. If a match is found between the component and the parent of a possible match, it would be advisable to try to find a match between the component and the parent's alternative children (i.e., siblings to the possible match). Therefore, in this situation the RESOLVER again would tighten the constraints on the possible match.

AKG compares, using the relaxation algorithm and the CKB, the validity of the system component connections. When a component is flagged as valid, AKG is then able to assign a function to it that is consistent with the target reasoning system. This approach to conflict resolution using the reasoning mechanism of constraint propagation raises the AKG system well above the capability level of a simple translator.

In summary, once the RESOLVER is called, all the components in the system are examined and the components with the highest confidence factors are marked. Based on the information (i.e., constraints) in these marked components, the propagation of confidence proceeds beginning with neighboring components. The propagation of confidence factors is global in the system and continues until all the components' confidence factors change less than some preassigned rate of convergence. At that time, the system's confidence factor is considered settled. The RESOLVER then scans all the components in the system and flags the components with confidence factors below the user-defined threshold. As a last resort, the RESOLVER asks the user to supply new information and confidence factors for these flagged components. This resolution process repeats until all the components' confidence factors exceed the threshold value.

3.0 TRANSLATION VERSUS INTELLIGENT INTERPRETATION

The following example identifies the difference between translation and interpretation using the conversion of a sentence from one language to another.

The original sentence (in Persian):

سرم سنگین

The literal (English) translation:

My head is heavy.

However, the correct interpretation of the sentence is:

I have a hangover.

The AKG system provides many advantages over a direct translation approach. A knowledge base translator is capable only of uncritically reformatting information explicit within its input data. An intelligent interpreter, however, is able to extend and correct input by inferring missing values and resolving conflicts. This ability is necessary for automated knowledge generation in the presence of sparse data such as that available from a CAD system.

The AKG prototype took as its testbed a demonstration circuit for purging pneumatic systems called the "Purge Demo." The knowledge base for this system had been manually constructed by NASA for verification using the Knowledge-Based Autonomous Test Engineer (KATE). Early work using a translator (Thomas 87) had indicated that translation is not sufficient for the resolution of CAD data into a knowledge base. A test of the AKG system was thus to autonomously produce a knowledge base which would closely approximate its human-generated counterpart.

The results of both studies are listed in Table 1. A description of the KATE slots depicted in the table may be found elsewhere [Cornell 87, Gonzalez et al. 88]. Note that the translation approach was found to be unable to provide any values for some KATE slots and it predicted a relatively low potential capability to fill others. In each of these cases the AKG intelligent interpretation approach is found to be superior. The component information of the CKB coupled with heuristic driven parsing will enable slots AN-ELEMENT-OF (AEO), TOLERANCE, DELAY, and STATUS (transfer function) to be filled at least 75% of time. It is estimated that the process information coupled with the TO-FROM list will allow identification of 90% to 100% of the SOURCE-PATH, IN-PATH-OF, SOURCE, and SINK slots.

	Translation Results			AKG Results		
Slots	Filled Auto.	Percent	Est. of Pot. Cap.	Filled Auto.	Percent	Est. of Pot. Cap.
aio	52/52	100%	100%	26/26*	100%	100%
aeo	NA	NA	NA	14/14*	100%	> 75%
nomencl.	0/52	0%	100%	13/13*	100%	100%
source-path	8/52	15%	50%	25/25#	100%	> 90%
in-path	52/52	100%	> 90%	35/35	100%	> 90%
source	2/2	100%	> 80%	NA	NA	100%
tolerance	NA	NA	NA	2/2*	100%	75%
delay	0/3	0%	0%	3/3*	100%	75%
status	0/52	0%	50%	6/6*	100%	75%
units	23/23	100%	100%	23/23	100%	100%
range	16/16	100%	100%	15/15	100%	100%
sinks	2/2	100%	80%	NA	NA	100%

Notes: (*) Filled with the help of the component database
(#) Need special operators to get this result.

Table 1. Comparison of Results.

4.0 CONCLUSIONS

This paper has discussed the structure and the operation of the Automated Knowledge Generator (AKG) system. It has been shown that a simple translator would not be sufficient to generate a viable knowledge base for a diagnostic system. An intelligent interpreter such as AKG is needed in order to accomplish the task of automatic knowledge acquisition from CAD databases. Work on the AKG system is continuing with work focussing on using CAD descriptions from a number of varied sources. These include Shuttle Ground Support subsystems, power generation systems, and Advanced Launch System processes.

ACKNOWLEDGEMENTS

The AKG research project is supported by NASA, Kennedy Space Center under contract NAG-10-0043.

REFERENCES

[Cornell 87] M. Cornell, "The KATE Shell - An implementation of Model-Based Control, Monitor and Diagnosis", Proceedings of the First Workshop on Space Operations Automation and Robotics, Houston, Texas, 1987.

[Gonzalez 88] A. J. Gonzalez, H. R. Myler, B. C. Owen, and M. Towhidnejad, "Automated Generation of Knowledge from CAD Design Data Bases", Proceedings of the First Florida Artificial Intelligence Research Symposium, Orlando, Florida, 1988.

[Gonzalez 89] A. J. Gonzalez, H. R. Myler, and M. Towhidnejad, "Automated Knowledge Generation in Support of Shuttle Ground Operations", Proceedings from the Artificial Intelligence in Government Conference, Washington, D.C., 1989.

[Kladke 89] R. Kladke, "A Mega-heuristic Approach to the Problem of Component Identification in Automated Knowledge Generation", Masters Thesis, University of Central Florida, November, 1989.

[Marcus 89] S. Marcus and J. McDermott, "SALT: A Knowledge Acquisition Language for Proposed-and-revised Systems" Artificial Intelligence, Vol. 39. No. 1. 1989. pp. 1-37.

[Rendell 85] L. Rendell, "Substantial Constructive Induction Using Layered Information Compression: Tractable Feature Formation in Search", Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Vol. 1., Los Angeles, CA, 1985.

[Thomas 87] S. J. Thomas, "Automated Construction of a Knowledge Base from Computer Aided Design Data", NASA Internal technical report, Artificial Intelligence Section, 1987.