A UNIFIED KNOWLEDGE BASE FOR REPRESENTING ELECTRICAL POWER SYSTEMS

Mohamed A. Madkour and Nagdy M. Nagdy
Systems and Computers Department, Faculty of Engineering,
Al-Azhar University

ABSTRACT

A unified knowledge-based methodology is introduced to develop expert systems that can handle a wide range of problems in managing electrical power systems. The developed approach provides a flexible way to deal with distribution networks and does not impose any restrictions. a GLOBAL knowledge base (GKB) is introduced to serve as an application-independent pool of knowledge. It would serve multiple distinct expert systems, each of which may be developed separately by adding an additional application-dependent knowledge module. The GBK is automatically generated by transforming a textual description of the entire power system into knowledge frames using a suitable parsing technique. A simple application module is developed to show the interfacing to the GBK.

Keywords: Knowledge-based systems, Electrical Power systems.

1. INTRODUCTION

Knowledge based techniques have great potential in handling several important problems that are encountered during the planning and operation of electrical power systems (EPS). Modern EPS's are complex large scale systems, and a typical system needs large numbers of highly experienced planners and operators to achieve proper management . However, since the numbers of such people are gradually decreasing there is an increasing need for expert systems which support system planning and operation and which can prevent human errors [1]. Examples for employing knowledge-based techniques in electrical power systems include transmission line protection [2], relay setting [3,4], substation supervision [5], and load allocation [6]. The list of such applications includes many others, and this situation gives rise to the need for a unified application-independent global knowledge base (GKB) that captures all relevant knowledge about the EPS on hand and serves multiple distinct applications.

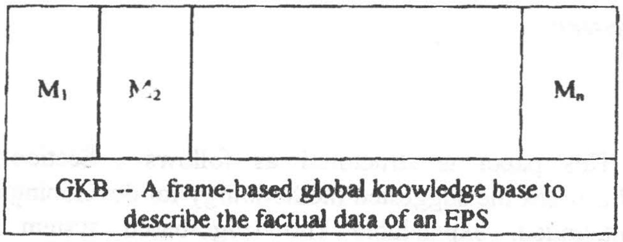
The concept of having an application-independent GKB is further developed in this paper. A prototype is implemented which integrates the GKB to an application-dependent expert system that handles the switching decision problem. Test results show that the developed expert system has very good performance which gives confidence in the feasibility of the suggested approach.

This paper is structured as follows . Section 2 illustrates the suggested methodology for developing an integrated, expandable knowledge-based system to manage electrical power systems. In section 3, the development and implementation of an application-independent global knowledge base is given. An automatic means for generating and updating the knowledge base is developed in section 4, and a sample application-dependent expert system is shown in section 5. Concluding remarks are given in the last section.

2. METHODOLOGY

Figure (1) outlines the suggested methodology for developing an integrated knowledge-based system to manage large scale EPS's. In this methodology, a unified global frame-based knowledge base is provided to hold all relevant facts for the entire distribution network including both its potential topology and the state of all devices and switchgear. This knowledge base would be viewed as an application-independent pool of common knowledge that serves multiple distinct expert system applications. Each expert system may be developed separately by adding an additional application-dependent module that may combine both rules and frames for representing specific knowledge, together with a suitable inference engine and user interface. The entire system has an open architecture which allows to develop and integrate new expert system modules as needed.

A flexible and automatic means should be provided for the initial generation and subsequent updating of the **GKB**. This is essential in view of the huge size of knowledge needed to represent actual large scale EPS's. A novel technique is adopted in the present work by using a special purpose symbolic network description language (called **GDL**) [7] to represent the entire electrical power system at the required level of detail using a plain text. By developing a proper GDL language analyzer, it would be possible to automatically transform a verified network description text into the equivalent **GKB** and save it in the required format on disk storage.



M_i = knowledge-based module for the i^th application

Figure 1. Conceptual view of the system integration.

The structure of the selected **GDL** language allows to analyze the text and store the **GKB** in a modular form in which there is a separate database for each substation and/or power plant in the power system. Consequently, any future changes in the network can be easily specified as a further **GDL** text and automatically transformed to update an existing **GKB**.

The entire knowledge-based system may be implemented using either a standard expert system shell or an AI language such as Prolog or Lisp. The later approach is considered in this paper to allow full control over the generation of the **GKB** and the internals of the expert system modules. Prolog is selected to make use of its internal backward chaining inference engine, as well as other reasons as outlined in [8].

3. THE KNOWLEDGE BASE

A power system is mainly composed of elements such as generators, loads, transmission lines, cables, transformers, busbars, switchgear, and etc. Such objects are modeled by frames that capture all types of available data in properly specified slots. for every individual object there should be a uniquely identifiable frame to model that object. A general form to represent frames in Prolog can be specified using two predicates. In this scheme, each individual frame would be represented by one *frame* predicate together with a number of *slot* predicates. These predicates are defined as follows:

frame(frame_pointer,identifying_slots(**location, numeral,partial,type,object_name**))
slot(frame_pointer,additional_slot(**slot_name**,slot_value))
where:

frame_pointer = a unique numeric pointer to identify a given frame within the GKB.
identifying_slots = a structure that contains a set of slots that uniquely identify an object.

additional_slot = a structure that contains a named slot.

The *frame* predicate maps the values of the objects identification slots into a unique numeric pointer value that is automatically assigned during the **GKB** generation. All additional slots of a given individual frame would be accessed through the same frame pointer value. As an example, Figure (2) shows a frame representation for a disconnecting switch.

slot: **location** value: SUB_1
slot: **numeral** value: 11/* Kilo Volt */
slot: **partial** value: BAY_1
slot: **type** value: disconnecting
slot: **object_name** value: BS_1
slot: **state** value: ON
slot: **reference** value: JUNCTION_1

(a) Pictorial view.

frame (123, identifying_slots(SUB_1,11, BAY_1, disconnecting, BS_1)).
slot (123, additional_slot(**state**, ON)).
slot (123, additional_slot(**reference**, JUNCTION_1)).

(b) Prolog Predicates, (assuming frame pointer = 123).

Figure 2. Frame representation for a disconnecting switch.

Slots should be capable to hold values taken from different domains such as integer, symbol, compound object, lists, and etc. A generalized Prolog domain declaration for the "slot_value" argument is given below to accommodate all required domains .

slot_value = domain_1 (argument_type_1); domain_2 (argument_type_2);... domain_N (argument_type_N)

4. AUTOMATIC KNOWLEDGE BASE GENERATION

A language analyzer is developed to map the knowledge representing the electrical power system from a GDL text into a frame-based knowledge base. Figure (3) outlines the steps for the GKB generation. The analyzer is fully implemented in Prolog, and it uses the augmented transition network (ATN) technique for parsing the GDL text [9].

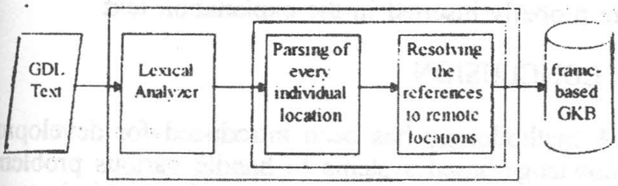


Figure 3. Automatic generation of the frame-based global knowledge base (GKB).

The physical storage of the GKB may be realized either by providing interface to a standard DBMS or alternatively by relying on the inherent storage facilities provided within Prolog. The second approach is considered to be an easy and fast solution. The entire GKB is specified in Prolog internal database predicates, and saved in a modular form on several files in standard Prolog internal database format. Each file contains a part of the GKB that represents a single location of the EPS.

4.1. *Lexical Analysis*

The lexical analyzer transforms the input text into a series of tokens. The EPS is logically divided into individual locations, and all tokens related to a given location are saved on a separate file using the Prolog external database facility. In this way , any changes to a given location can be incorporated easily by editing the original GDL text for only that location and generating the corresponding modified external database file without affecting the existing files of the other locations. In the lexical analysis stage, the Prolog's external database files are more suitable than its internal database to allow processing of large scale EPS description on disk storage .

4.2. *Parsing*

The developed parser transforms the output of the lexical analyzer into a frame-based GKB that represents the entire EPS. Numeric frame pointers are automatically generated during parsing and uniquely assigned to the different frames. Parsing should take place in two steps as shown in Figure (3) to permit resolving the problem of unknown frame pointers when generating frames for transmission lines that connect remote locations. In the first step, parsing of every individual location of the EPS would be done straight forward. The parser transforms the Prolog external database file of tokens of a given location (generated during lexical analysis) into Prolog internal database files containing the frames for all objects inside that location. However, frames that represent transmission lines which interconnect remote locations cannot be generated in a single step, and their generation should be delayed. After parsing all locations, every frame in the GKB would have its own unique frame pointer value, and this permits to generate the transmission lines' frames.

5. AN EXPERT SYSTEM EXAMPLE

This section gives an example for developing an expert system module to handle the switching decision problem according to the structure of Figure (1). In addition to the unified application-independent GKB, the module includes an application-dependent knowledge base, a backward chaining inference engine and a user interface. The developed expert system is implemented in Prolog and integrated with the GKB.

5.1. *The Switching Decision Problem*

Switching decision is a vital real time application in EPS's. Daily operation in substations involves switching acts upon circuit breakers, disconnectors, earthing switches, and etc.. Both load transfer in normal conditions , and fault clearing in emergency cases are two examples for real life tasks that require taking fast switching decisions based on strict interlocking rules. Any human mistake in a switching operation would compromise the safety of both the electrical equipment and the operation personnel. Consecutive computer-aided methods have been published previously [10-12] for distribution network interlocking based on a topological representation of the EPS. However, knowledge-based techniques for interlocking would be more effective and flexible in handling the full spread of switching decisions. The expert system is required to answer questions whether a specified switching device can be operated (opened or closed). Answers to such questions are based on the topological connections, the current states of the relevant switching devices, and the interlocking rules.

5.2. *Interlocking Knowledge*

Interlocking is classified into three categories based on the functional characteristics of the switching devices. The basic interlocking knowledge for these categories are summarized below in a rule style.

i- circuit braking devices:
- A circuit breaking device can be **opened** unconditionally.
- A circuit breaking device can be **closed**
  if the path of the breaker's current does not contain any earth connection.

ii- disconnecting devices:
- A disconnecting device can be **opened** and/or **closed**
  if the disconnector's current will not be switched.
- The disconnector's current will **be switched**
  if the current's path is not opened in front of the disconnector,
  and the current's path is not opened behind the disconnector,
  and the disconnector does not have an external impedance free path across its terminals.

iii- earthing devices:
- An earthing device can be **opened** unconditionally.
- An earthing device can be **closed**
  if the path is opened in front of the earthing device,
  and the path is opened behind the earthing device.

By accessing the frame-based **GKB**, which contains all topological details and switches' status information, the expert system module can find out the proper switching decision. Backward chaining is most suitable for obtaining a decision whether to permit a specified switching action. Figure (4) gives an example for an inference tree based on the interlocking rules for a disconnecting device.

5.3. *Case Study*

As a vehicle to show how the developed expert system provides switching decisions, a simple example of two connected identical substations is considered. Figure (5) shows the single-line diagram for the selected example.

Three sample cases are considered to give examples for applying the interlocking rules of the three switching categories. A summary of the switching decisions and their explanations, as obtained by the expert system, is shown in Table (1) based on the switch's states depicted in Figure (5). By remembering the trace of used interlocking rules in a given case, the system provides pre-stored (canned) explanations, with names for both switches and bays as parameters that are properly inserted in the explanation text.

6. CONCLUSION

A methodology has been introduced for developing knowledge-based systems to handle various problems in planning and managing large scale electrical power systems. The basic concept is to build a global application-independent knowledge base that captures all factual knowledge about the topological connections and actual states of all components in the electrical power system. Several application-dependent expert system modules may be developed individually to handle specific problems. These modules would be integrated with the global knowledge base which acts as a pool of common knowledge. This methodology provides an open structure in which there is no conceptual limit on the number of expert system modules that can be developed and added to the entire system.

A novel technique is presented for building the global knowledge base (GKB).A symbolic description language is used to specify the entire electrical power system in a plain text, and then automatically transforming the text into the required **GKB**. Any future changes in an existing electrical power system can be specified as a further descriptive text which would be automatically transformed and integrated with the old GKB to update its contents.

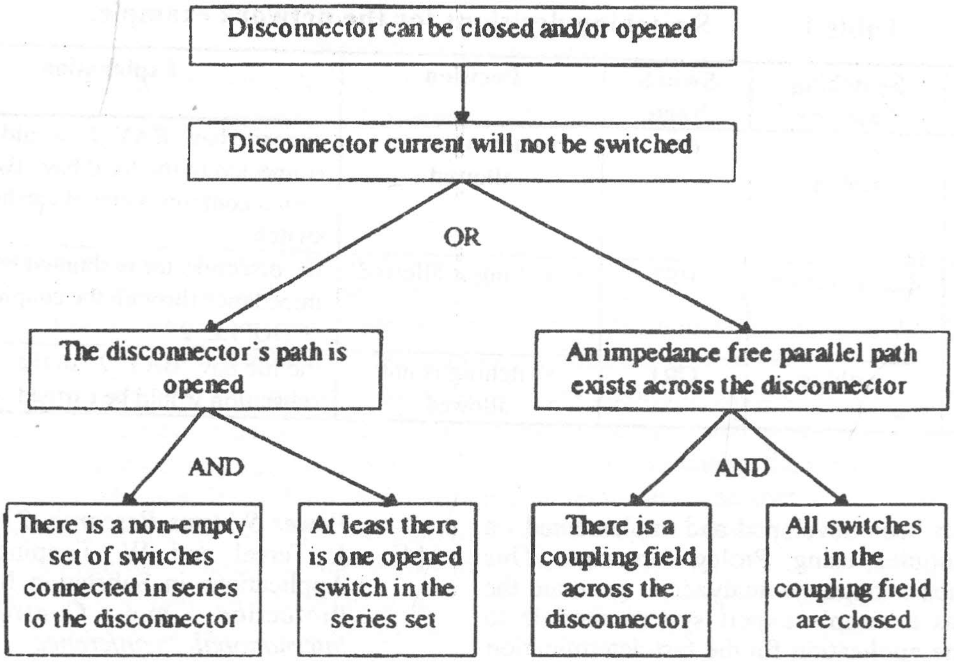


Figure 4. Inference tree for the disconnector's interlocking rules.

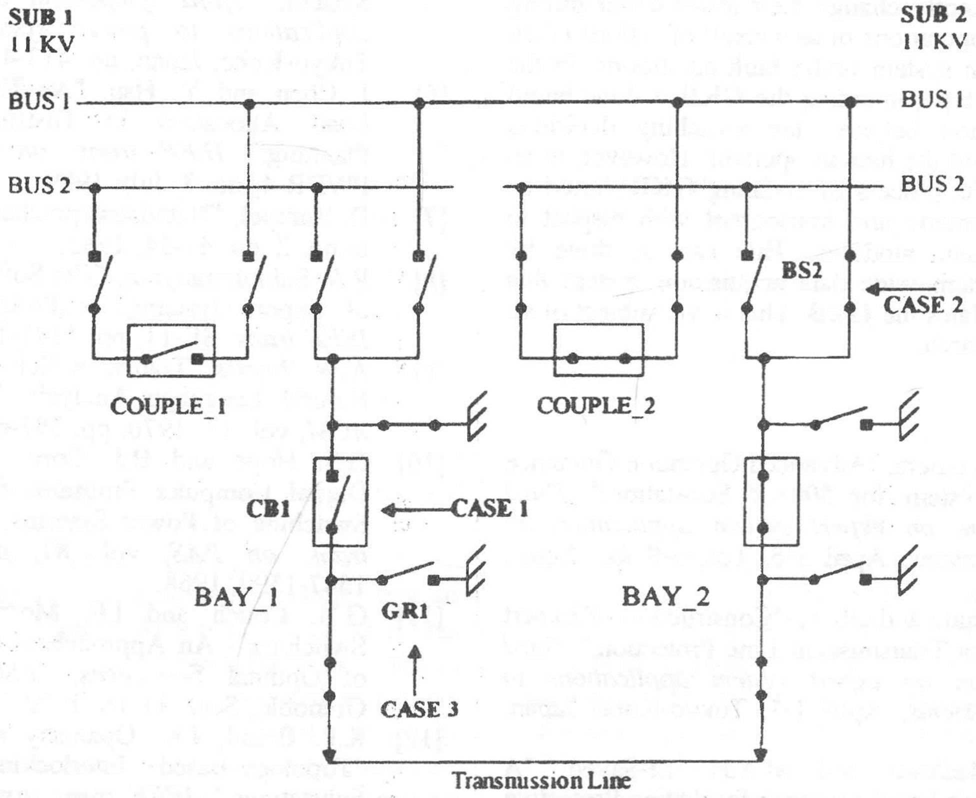


Figure 5. Example for two connected substations.

Table 1 Switching decisions for the network example.

Case	Switching Category	Switch Name	Decision	Explanation
1.	circuit breaking	CB1	switching is not allowed	opposite bay "BAY_2" would be connected to the local bay "BAY_1" which contains a closed earthing switch
2.	disconnecting	BS2	switching is allowed	the disconnector is shunted by zero impedance through the couple bay "COUPLE_2."
3.	earthing	GR1	switching is not allowed	the life bay "BAY_2" in the opposite substation would be earthed.

A prototype has been developed and implemented on a personal computer using Prolog language. This prototype includes a language analyzer to generate the **GKB** as well as a sample expert system module to handle a real-time application for the fast determination of switching decisions.

The process of continuously updating the **GKB** needs further elaboration. Switches in a real electric power system do frequently change their states either during normal routine operations or as a result of actions taken by the protection system under fault conditions. In the developed prototype, updating the **GKB** is done based on the interaction between the switching decisions expert system and the human operator. However, in an actual system, the process of updating **GKB** should be completely automatic and transparent with respect to the expert system modules. This can be done by providing a system wide data acquisition system that periodically updates the **GKB**. This is the subject of an undergoing research.

REFERENCES

[1] S. Ito and others, "Advanced Operation Guidance Expert System for 500KV Substation," *Third symposium on expert system applications to power systems*, April 1-5, Tokyo-Kobe, Japan, 1991.

[2] K. Kawahara and others, "Construction of Expert System for Transmission Line Protection," *Third symposium on expert system applications to power systems*, April 1-5, Tokyo-Kobe, Japan, 1991.

[3] M.A. Madkour and M.A.H. El-Sayed, "A Knowledge-based approach for Setting Protective Relays in Transmission Networks," *Electric Power Systems Research*, 27, pp.107-115, 1993.

[4] J. Tercal and W. Jianping, "Expert System Applications in Substation Monitoring, Back-up Protection, and Control," *IEE fourth international conference in power system protection development*, April 11-13, U.K, 1989.

[5] S. Muto, T. Matsuda and Y. Yamagata, "Supervisory System for Substations with Expert System," *Third symposium on expert system applications to power systems*, April 1-5, Tokyo-Kobe, Japan, pp. 413-418, 1991.

[6] J. Chen and Y. Hsu, "An Expert System for Load Allocation in Distributed Expansion Planning," *IEEE trans. on Power Delivery*, PWDR-4, no. 3, July 1989.

[7] D. Rumpel, "Netzdatensprache." *etz-Archiv*, vol. 5, no. 2, pp. 47-54, 1982.

[8] P.A. Subrahmanyam, "The Software Engineering of Expert Systems: Is Prolog Appropriate?," *IEEE trans.* SE-11, pp. 1391-1400, 1985.

[9] A.W. Woods, "Transition Network Grammars for Natural Language Analysis, "*Commun. of the ACM*, vol. 13, 1970, pp. 591-606.

[10] G.S. Hope and B.J. Cory. "Development of Digital Computer Programs for the Automatic Switching of Power Systems Networks," *IEEE trans. on PAS*, vol. 87, no. 7, July, pp. 1587-1599, 1968.

[11] G.H. Couch and I.F. Morrison. "Substation Switching - An Approach to the Determination of Optimal Sequences," *PSCC Proc.* V. II, Grenoble, Sept. 11-16, 1972.

[12] K.P. Brand, J.K. Opainsky and W. Wimmer. "Topology-Based Interlocking of Electrical Substations," *IEEE trans. on Power Delivery*, PWRD-1, no. 3, July 1986, pp. 118-126.