

# NETMAT: a knowledge-based grid system analysis tool

S Zitouni and M R Irving

Brunel Institute of Power Systems, Brunel University,  
Uxbridge UB8 3PH, UK

*The increasing expansion of electric power systems renders the power system operator's task increasingly complex. The integration into energy management systems of further analytical algorithms implies that more data has to be analysed by the control engineer. For these reasons and many others, more sophisticated tools are required by power engineers to ease the pressure under which they perform their task. The advent of knowledge-based systems has led to a new approach to the problem. The combination of expert systems and numerical algorithms can be advantageously exploited to assist the power system engineer in operating the system. This paper presents the development of a knowledge-based tool for grid system analysis. The tool, NETMAT (NETwork Modelling AssistanT), is to be used to analyse the impact of grid system maintenance and modification procedures and of new generating plants on power utilities, and in particular on their ability to generate and sell electricity. NETMAT consists of a number of numerical applications interfaced to an expert system shell through specific problem domain knowledge bases. Results are presented based on the use of the IEEE-30 busbar network as a test network.*

*Keywords: a.c. load flow, sensitivity analysis, transient stability analysis, fault levels, expert systems, object-oriented language*

## I. Introduction

Since their introduction in the electric power system engineering field, a knowledge-based systems have been increasingly accepted and used by power supply utilities in a wide range of applications ranging from on-line training tools to integration in energy management systems<sup>1-3</sup>. This interest can be explained by the many features that knowledge-based systems provide and which are not usually available from conventional programs. Among these features the following can be cited:

(1) Unlike many conventional programs, knowledge-

based systems provide a flexible user friendly interface enabling easy modification of the data used by numerical algorithms. Knowledge-based systems also have an explanatory facility.

- (2) The reasoning of expert systems is more acceptable to the operator/planner than the solution of numerical algorithms, as such reasoning matches human cognitive processes
- (3) Knowledge-based systems address problems where the knowledge may be less deterministic, more imprecise and requiring reasoning under uncertainty, whereas conventional programs deal with problems where mathematical models are available.
- (4) There is a greater possibility of revision, maintenance and evolution of the system by non-programmers after installation.

The features stated above show that knowledge-based systems complement rather than replace or compete with numerical algorithms. Therefore, the combination of both techniques enables the creation of a flexible computing environment for power system engineers.

## II. Expert system integration

The increasing expansion of electric power systems renders the power system operator's task more complex. The integration into energy management systems of further analytical algorithms implies that more data has to be analysed by the control engineer. For these reasons and many others, more sophisticated tools are required by power system engineers to ease the pressure under which they perform their task. The advent of expert system technology has led to a new approach to the problem, with the integration of knowledge bases which are friendly to the user, enabling easy modifications to the data required by numerical algorithms in the light of new decisions. The power system analyst's tools such as load-flow analysis, security assessment, alarm processing and stability analysis are excellent candidates for this approach. Changes to the topology of the network, to its loading, and to its constraints can all be interfaced through a knowledge-based system, which has the ability to give decision support to the user. The integration of expert systems with numerical algorithms can provide the

powerful operational tool which is required by electric power utilities<sup>4</sup>.

In the following sections, the development of a knowledge-based tool for grid system analysis will be presented. The tool, NETMAT, is used to analyse the impact of grid system maintenance and modification procedures and of new generating plants on power utilities, and in particular on their ability to generate and sell electricity. In the first phase of the project, a working prototype version was developed with particular effort given to the numerical algorithms, to the choice of the expert system shell and to the design of the interface between the algorithms and the shell. In a second phase, special attention will be given to the inclusion of additional human expertise into knowledge bases which have already been developed and the interrelationships between the various knowledge bases.

### III. Functional specifications

The knowledge-based system consists of knowledge based modules and FORTRAN application modules. The knowledge bases include:

- (1) the system manager knowledge base, this is a common knowledge base which enables the user automatically to load, consult and unload any specific problem domain knowledge base;
- (2) the sensitivity test knowledge base;
- (3) the fault level knowledge base;
- (4) the transient stability knowledge base.

The FORTRAN applications include:

- (1) the sensitivity test program;
- (2) the fault level program;
- (3) the transient stability studies program;
- (4) the AC load program<sup>5</sup>.

Figure 1 represents the structure of the developed system.

### IV. System implementation

The system has been implemented on an 80486 DX2-66 MHz IBM personal computer with expanded memory and large capacity hard disc to handle very large data-bases. The development tool is the commercial expert system shell. *Nexpert Object*<sup>6</sup>. This shell is recognised as one of the leading expert system tools, and has an installed base of more than 14 000 systems worldwide.

The Microsoft FORTRAN compiler<sup>7</sup> (DOS and Windows version) is used to compile and link the numerical applications such as the AC load flow and the transient stability program.

### V. Nexpert object

*Nexpert Object* is an expert system shell from Neuron Data Corporation. It provides object-oriented data representation capabilities with classes, objects, slots and metaslots. Classes and objects are defined as shown in the following example

```
(@CLASS = lines
  (@SUBCLASS =
    overloaded_lines
  )
  (@PROPERTIES =
    name
    node1
    node2
    loading
  )
)
(@OBJECT = RecordNumber
  (@PROPERTIES =
    Value @TYPE = Integer;
  )
)
```

Dynamic objects can be created and deleted at run time. Objects can inherit properties from other objects or

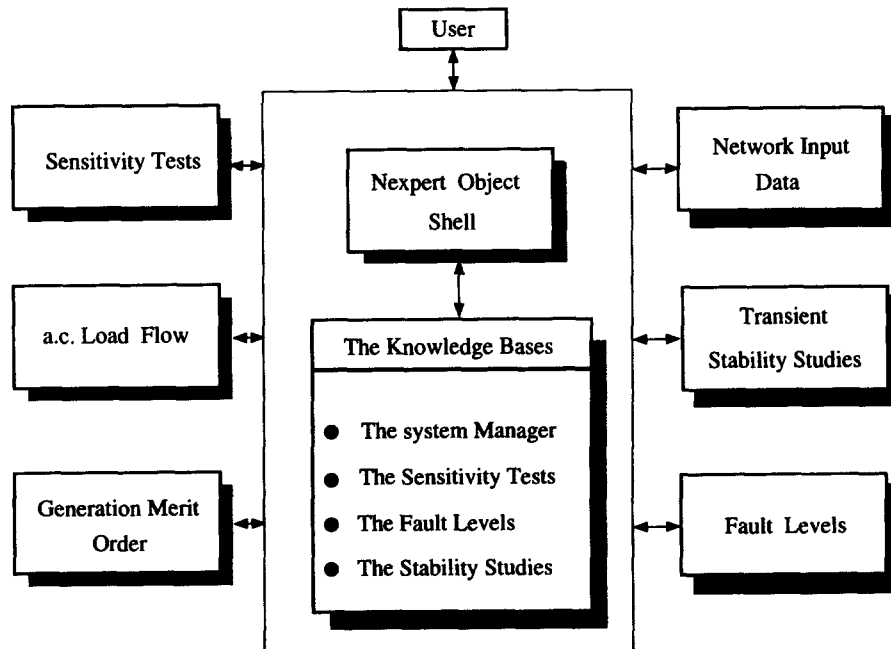


Figure 1. Structure of the knowledge-based system

classes in a variety of ways. Inheritance and inference strategies can be changed at any time, during processing, by using some built-in functions. *Nexpert Object* uses four inference search mechanisms with varying priorities: backward chaining, forward chaining, semantic gates and context links. These search mechanisms are controlled by the knowledge-based system according to the inference strategy. They can also be globally or locally disabled or limited.

The expert system shell is the core of the expert system. The power of its inference engine, its user friendliness as well as its interfacing facility to external applications are the dominant factors in making the knowledge-based system a powerful and effective tool. *Nexpert Object* is a state-of-the-art shell that is widely used. In the USA, the electric power industry has been using this tool<sup>8</sup> as early as 1989. It is portable to many hardware platforms (VAX, SUN, Macintosh, PC, etc.). *Nexpert Object* features a powerful graphical development environment and has an open architecture that allows integration with industry-standard databases, graphical systems and many other applications. In choosing this shell, the possibility for the system to be interfaced with graphics packages and relational data-bases was also taken into consideration.

The decision to use the expert system shell *Nexpert Object* instead of a conventional programming language such as FORTRAN was dictated by the following reasons.

- (1) Conventional programs do not match the human heuristic thought process.
- (2) In *Nexpert Object*, the knowledge required by the operator to support his/her decision making is explicit in the program coding.
- (3) Conventional programs do not encourage interaction with the user while they are running.
- (4) *Nexpert Object* improves the user friendliness of conventional programs.
- (5) The use of the *Nexpert Object* shell reduces the development time.

- (6) *Nexpert Object* allows the inclusion of the human expertise which cannot be coded in a conventional program (i.e. expertise which is less deterministic, more imprecise and has some degree of uncertainty).
- (7) *Nexpert Object* has explanatory facilities to explain what is happening in the program.
- (8) *Nexpert Object* runs on various platforms, making it easy to move the developed system from a PC to a UNIX/VMS workstation.

## VI. Knowledge bases

When building the various knowledge bases, a hierarchical and modular knowledge base architecture was adopted (Figure 2). We believe that this type of architecture will keep the expert system clear and easy to maintain in the future. It will also enhance the control of the expert system inference engine. The following knowledge bases were developed:

- (1) a central knowledge base which is the system manager;
- (2) the sensitivity tests knowledge base;
- (3) the fault level knowledge base;
- (4) the transient stability studies knowledge base.

The system manager is a common knowledge base which enables the user automatically to load, consult and unload any specific knowledge base. The other knowledge bases are specific problem domain knowledge bases. Each knowledge base calls and runs the corresponding FORTRAN application, retrieves the results, processes them and then presents the appropriate conclusions to the user.

### VI.1 System manager

The system manager knowledge base provides the user with the choice of consulting any of the existing knowledge bases without leaving the application. Different applications are loaded, consulted and then unloaded by the system manager. The system manager is composed

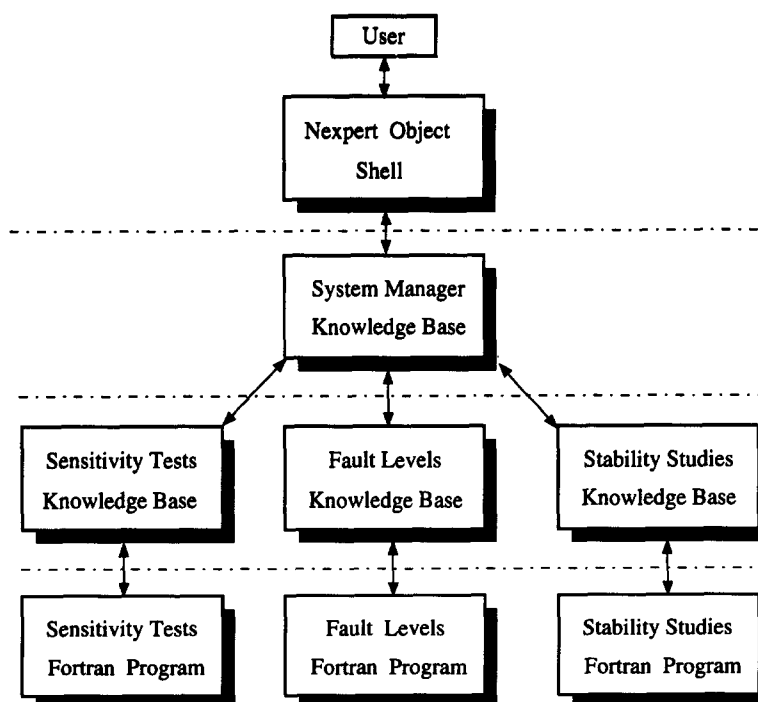


Figure 2. Hierarchical structure of the knowledge-based system

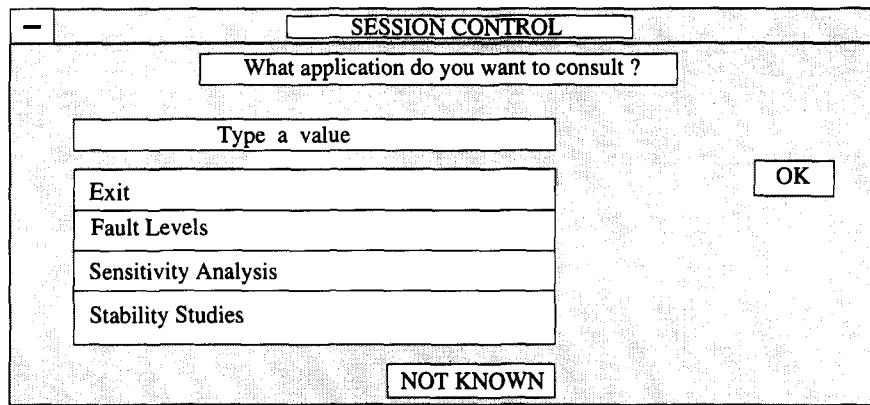


Figure 3. The system manager window

of a master rule and a set of slave rules. The master rule always checks whether there is something to be done; if not the session ends. If the user wants to consult an application, the slave rules load, process and unload the selected knowledge base. At the end of each consultation, control is given back to the master rule. Figure 3 represents the system manager window.

#### Master Rule

```
(@LHS =
  (Yes (Load_Application))
  (Yes (Get_User_Choice))
  (IsNot (user_choice) ("Exit"))
)
(@HYPO = Consult_Applications)
(@RHS =
  (Reset (application))
  (Reset (user_choice))
  (Reset (Load_Application))
  (Reset (Get_User_Choice))
  (Do (Load_Application) (Load_Application))
  (Do (Get_User_Choice) (Get_User_Choice))
)
```

#### Slave Rule

```
(@LHS =
  (Name ("c:\kbases\comkb") (path))
  (Is (application) ("stability_studies"))
)
(@HYPO = Load_Application)
(@RHS =
  (LoadKB ("@V(path)\stabilit.tkb") (@LEVEL = ENABLE;))
  (Do (Stability_Studies) (Stability_Studies))
)
```

#### VI.2 Sensitivity tests

The sensitivity tests application is aimed at determining the impact of circuit overloads on generating plants of interest. It enables the operator to find out the sensitivity of the system to circuit outages and changes in generation pattern. It also prompts the user to examine any significant circuit overloads and violations of voltage operating limits. The sensitivity tests knowledge base is the main knowledge base (Figure 4). First, the contingency outage data is checked against the maintenance outage data. Then, any circuit overloading due to maintenance outages is displayed to the user. For each circuit overloading due to a contingency outage, the sensitivity of the

overload to the output of pre-specified generating plants is checked, and the user is warned of any significant sensitivities.

#### VI.3 Fault level calculations

At this stage of development, the fault level application is a basic knowledge base which executes the fault level

program and uses the results to check whether the fault level at each busbar of the system is violating the circuit breaker rating. In case of violations, NETMAT warns the user about these violations, their degree as well as their location on the network. In this knowledge base, a grouped retrieve is used to access the FORTRAN results. In a grouped retrieve method, all records are retrieved in one operation and objects are dynamically created to hold the retrieved data.

#### VI.4 Transient stability studies

The transient stability application enables the operator to check the stability of specific generators

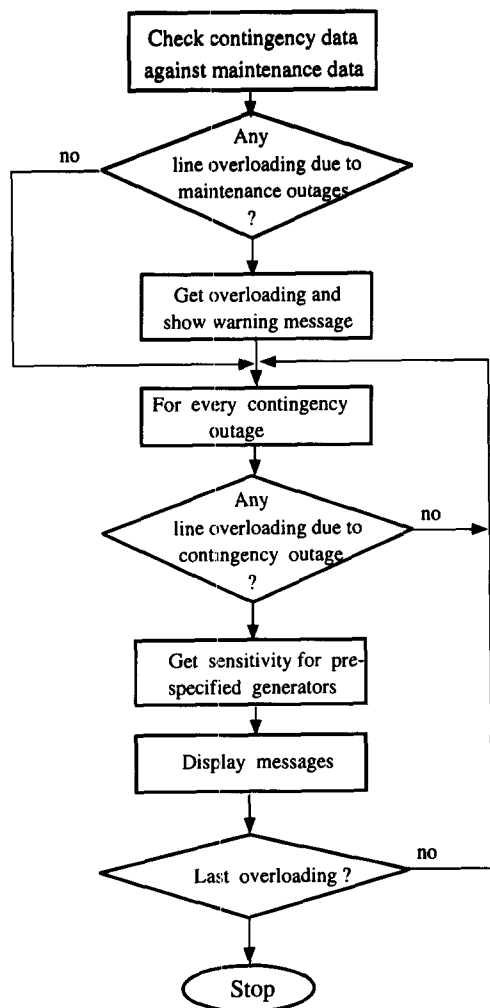


Figure 4. Flow diagram of the sensitivity tests knowledge base

following a close-up or a remote fault of finite duration.

The transient stability knowledge base invokes the transient stability FORTRAN program, retrieves its results and then checks the stability of the generators. When retrieving the FORTRAN results, NETMAT uses a sequential retrieve method. In this method, one record is retrieved at a time. The knowledge base processes that record before retrieving the next. A pointer is used to keep track of the next record to be retrieved and processed.

## VII. Numerical algorithms

### VII.1 Generation merit order

The generation merit order is used to determine the generators' contribution to meeting the total demand and the transmission losses according to their cost of generation or generation bidding price for pool systems.

The merit order method consists of minimizing the function

$$C_1X_1 + C_2X_2 + \dots + C_kX_k$$

where  $k$  is the number of generators,  $X_1, X_2, \dots, X_k$  are the generations, and  $C_1, C_2, \dots, C_k$  are the generation costs or bid prices.

subject to

$$X_1 + X_2 + \dots + X_k + losses = X_{total} \quad (1)$$

and

$$X_{1l} < X_1 < X_{1u}$$

and

$$X_{2l} < X_2 < X_{2u}$$

.....

and

$$X_{kl} < X_k < X_{ku}$$

where  $X_{il}$  and  $X_{iu}$  are the lower and upper generation limit for generator  $i$ .

Initially, the generations are set to their lower limits  $X_{1l}, X_{2l}, \dots, X_{kl}$ . If equation (1) is met, then the problem is solved, otherwise, the generator  $i$  with the least cost is chosen and  $X_i$  is set to its upper limit  $X_{iu}$ . If  $X_1 + X_2 + \dots + X_k + losses > X_{total}$ , then  $X_i$  is reduced till equation (1) is met. The generations  $X_1, X_2, \dots, X_k$  are then used as input data by the load flow and the sensitivity tests programs.

### VII.2 Sensitivity tests

The sensitivity tests program first checks that the maintenance outages are not causing overloadings on the system. Then, for each contingency outage and for each circuit overloading the sensitivity of pre-specified generators is calculated.

The sensitivity is defined as:

$$sensitivity = \frac{\text{change in overload}}{\text{change in generation}}$$

A measure of overload severity has been adopted. This measure is defined as follows.

- (1) The line is said to be near to the overload limit if the loading is between 90% and 100% of the overload limit.
- (2) The line is slightly overloaded if the loading is between 100% and 110% of the overload limit.
- (3) The line is significantly overloaded if the line loading is between 110% and 130% of the overload limit.
- (4) The line is severely overloaded if the loading is between 130% and 160% of the overload limit.
- (5) There is an extremely severe overloading on the line if the loading is greater than 160% of the overload limit.

Figure 5 shows the flow diagram of the sensitivity test program.

### VII.3 Transient stability studies

The transient stability program is based on the 'equal area criterion' (Figure 6). It determines a power boundary for a generator or group of generators in case of a fault of finite duration. The calculations consist of forming system impedances, deriving system admittances and varying voltages and transformer tap ratio within user defined limits. For each fixed set of the above parameters, the generated power is varied and the stability of the generator set tested by integrating the swing equation (2).

$$\frac{H}{50\pi} \frac{d^2\delta}{dt^2} = P_m - P_g \quad (2)$$

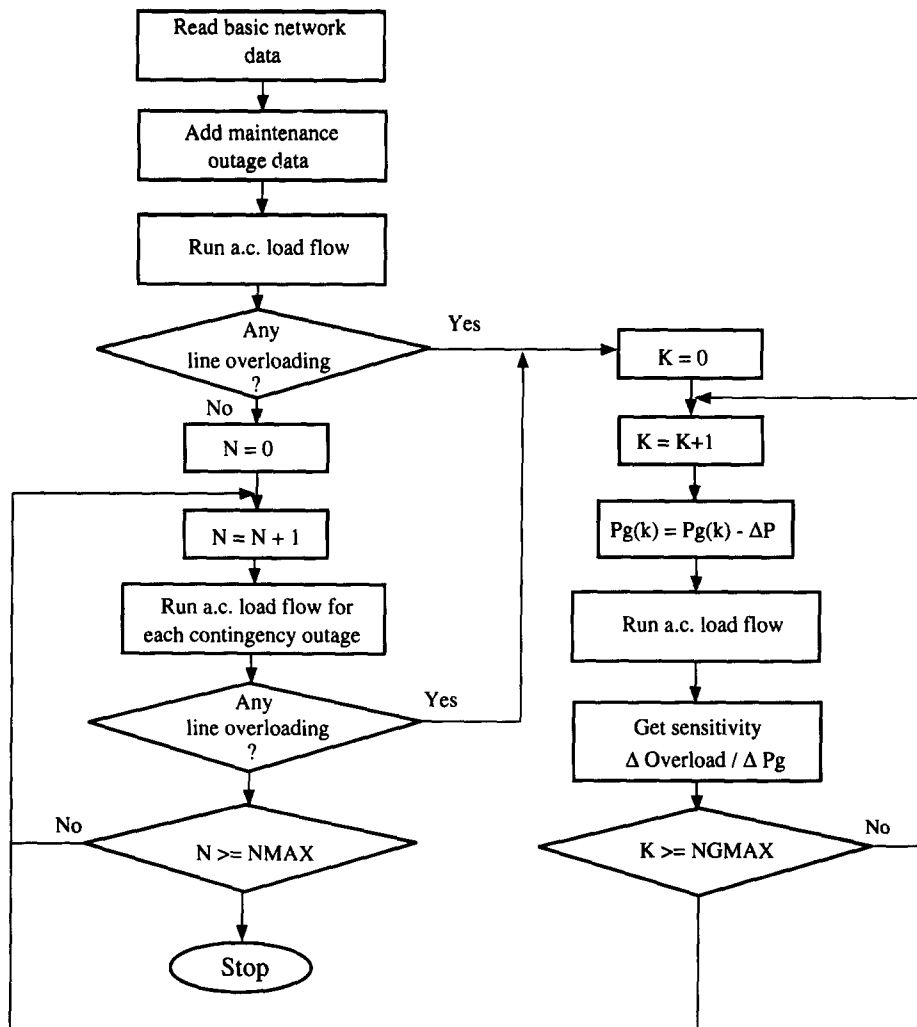


Figure 5. Flow diagram of the sensitivity tests program

where

$H$  inertia of the machine

$\delta$  rotor angle

$P_m$  mechanical power supplied to the generator

$P_g$  generated power.

The limitation of this program is that it finds the stability limit for the machine or machine group under the assumption that the remainder of the system may be represented by an infinite busbar (Figure 7) where  $Z'_d$  machine impedance

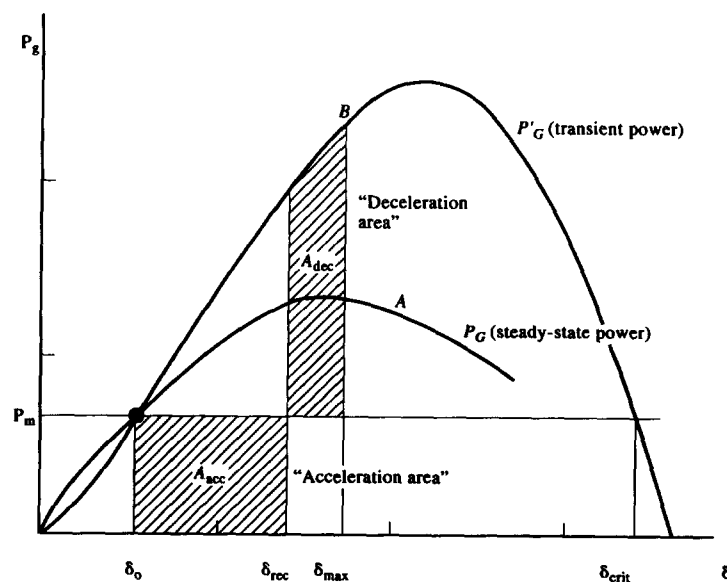


Figure 6. The equal-area criterion

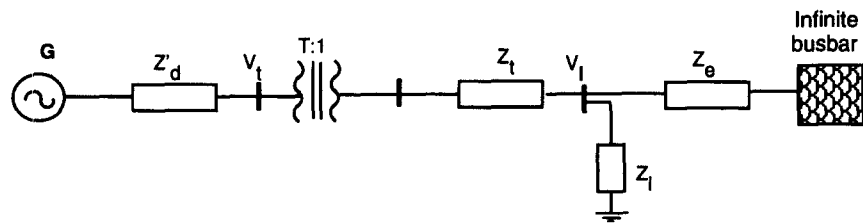


Figure 7. Simple equivalent model of a power system

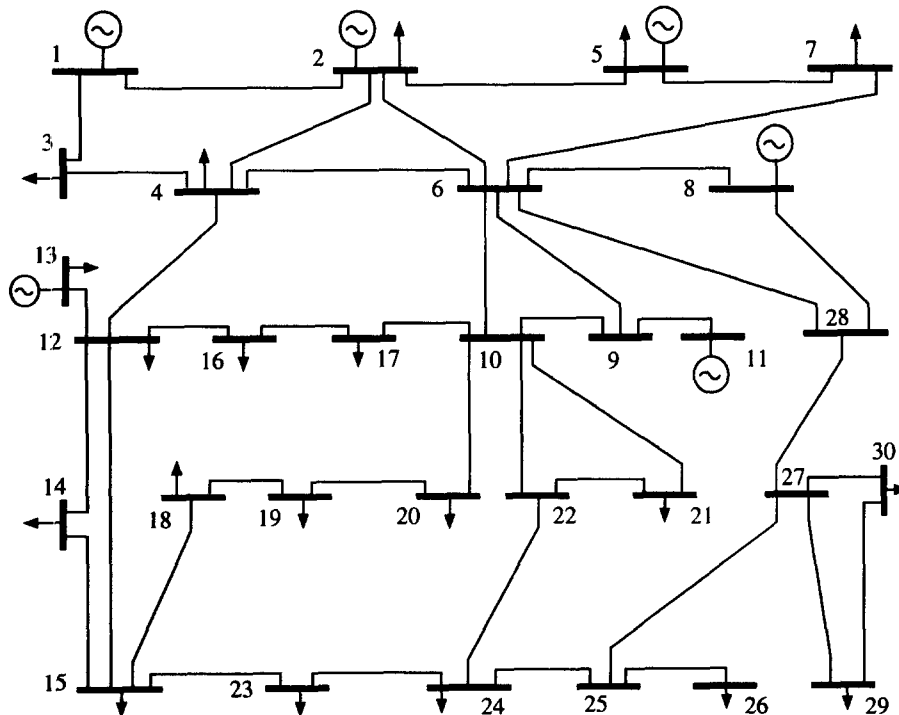


Figure 8. The IEEE 30-Busbar network

- $E'$  voltage behind the machine impedance
- $V_1$  terminal voltage
- $Z_t$  transformer impedance
- $V_1$  load voltage
- $Z$  load impedance
- $Z_e$  impedance representing the infinite bus
- $V_s$  infinite bus voltage.

have the following format:

```

field1| field2| field3| field4|
*****|
val11| val12| val13| val14|
val21| val22| val23| val24|
...
*****
    
```

**VIII. Tests and case study example**

The system has been successfully tested using the IEEE 30 busbar network (Figure 8). The input data for this network is given by Reference 9. The numerical applications generate output files which are used by the knowledge bases during the processing of rules. These files must have one of the spreadsheet or database formats that are supported by Nexpert database Bridges. In our case, Nexpert flat-file format (.NXPDB) was used. These files

NXPDB files are ASCII files and thus can be ported from one machine to another. The first two lines are the file header and the last line of asterisks (\*) indicates the end of the file. Every line between the second and the last line represents a record. Table 1(a) and (b) give some examples of files which are generated by the sensitivity tests program.

In Nexpert Object the processing of a knowledge base is started by suggesting hypotheses or volunteering data

**Table 1. Examples of NXPDB files**

(a)	(b)
LNNAME OUTAGE SNODE RNODE OVERLOAD	GENNAME GENNODE CIRCUIT DELTA
*****	*****
LINE32 LINE21 NOD23 NOD24  0.9750	GENER1  NOD2  LINE32 0.9281
*****	GENER2  NOD8  LINE32 0.9468
*****	*****

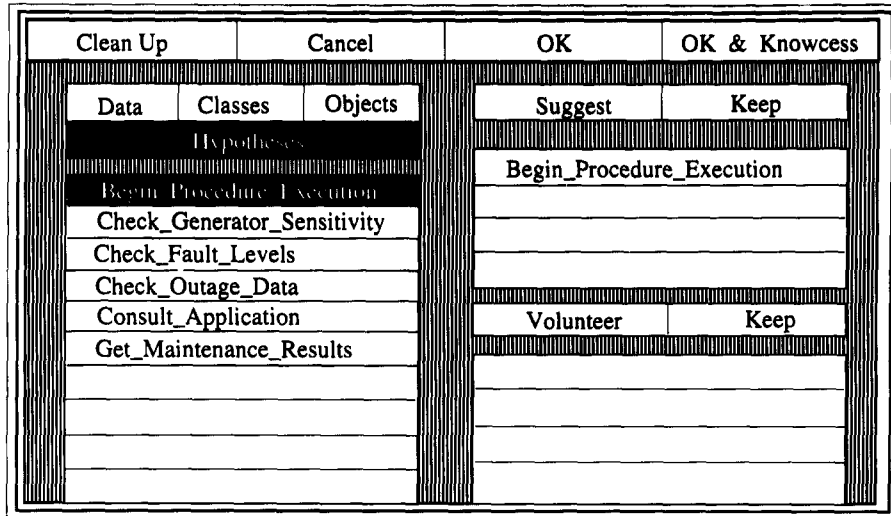


Figure 9. Suggest/volunteer dialog window

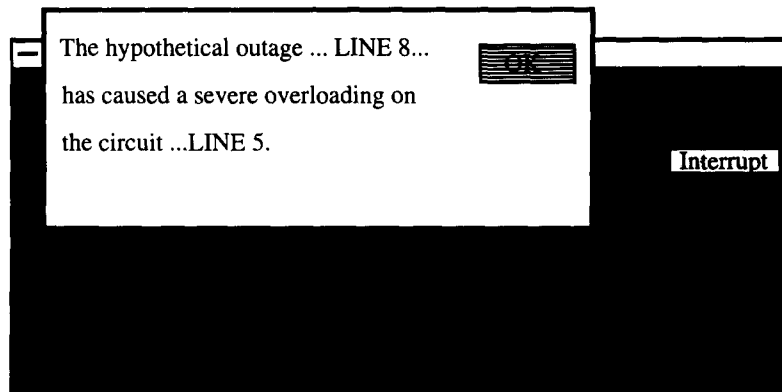


Figure 10. A NETMAT warning message

to the agenda. The agenda is a dynamic mechanism used by *Nexpert Object* to schedule events according to their priorities. Figure 9 depicts a *Suggest/volunteer* dialog window.

In NETMAT, however, the application is loaded and processed using the session manager knowledge

base. As an example, let us consider the sensitivity test knowledge base. Using the session manager (Figure 3), the knowledge base is loaded and the hypothesis *Begin\_Procedure\_Execution* is automatically put on the agenda. After retrieving the FORTRAN program results (Table 1), the knowledge base processes these results

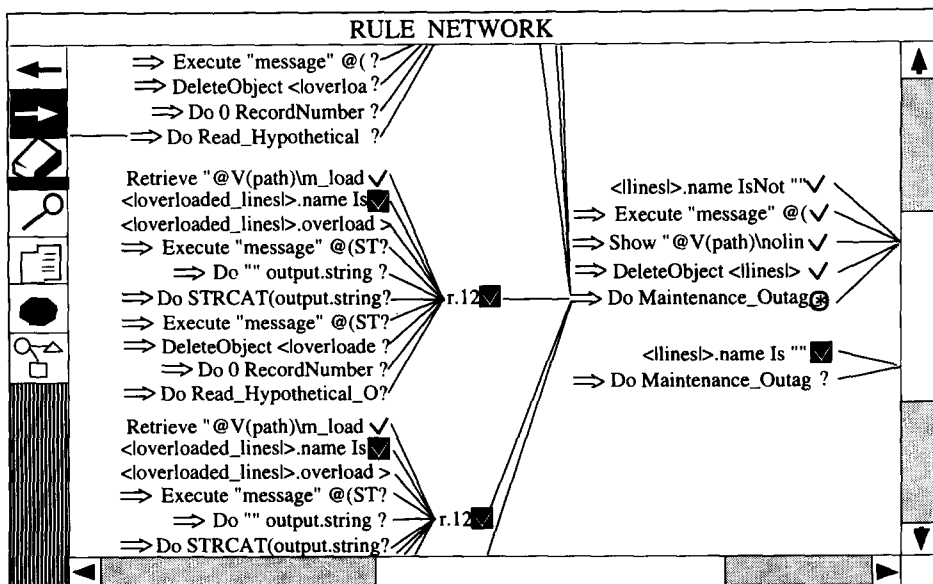


Figure 11. Rule network window



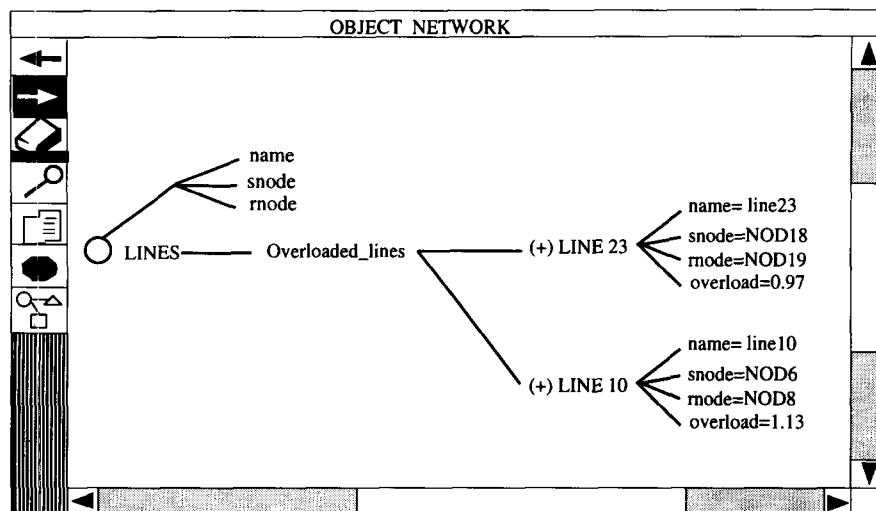


Figure 12. Object network window

and then warns the user of any abnormal condition on the system. The message depicted in Figure 10 is a typical warning message displayed by NETMAT.

During the knowledge base consultation, the user may stop the session and check the reasoning pathways of the knowledge base by viewing the rule network (Figure 11). The rule network shows the rule in a hierarchical network with the current status for each rule clearly indicated. Initially, all rules and hypotheses are preceded by a question mark, meaning that no hypothesis or rule is under evaluation. If a rule (hypothesis or condition/action) is being evaluated, it is preceded by an asterisk (\*) and is presented in a predefined colour. If the rule has been evaluated, it is preceded by a tick and is given another predefined colour. As *Nexpert Object* reasons on both rules and objects, an object network can also be viewed (Figure 12). The network shows the object knowledge structures in a hierarchy with the properties propagated over the inheritance pathways as well as their current values. At the end of each session, a full report giving detailed information about the session and/or a case status report which summarizes the session is available to the user. The entire session can be recorded in a

journal file if the user requires a permanent record of the consultation. Figure 13 shows a full report of a session on sensitivity tests.

## IX. Conclusions

In this paper, a knowledge-based system for grid system analysis has been presented. The tool, NETMAT, will be used to assist power system engineers in every day system operations and for training novice engineers. NETMAT is a user-friendly system which uses the user interface and the inferencing power of the expert system toolkit *Nexpert Object* to reason on objects and rules. This work has shown how the use of expert systems as a front end to numerical programs may assist power system engineers and other analysts in their decision making. The engineer's task is simplified by reducing the amount of data to be processed before taking any action. The combination of numerical algorithms with expert systems allows the inclusion of expertise and rules of thumb used by senior engineers, hence providing a mechanism for disseminating knowledge about the system to other engineers. The use of an object-oriented

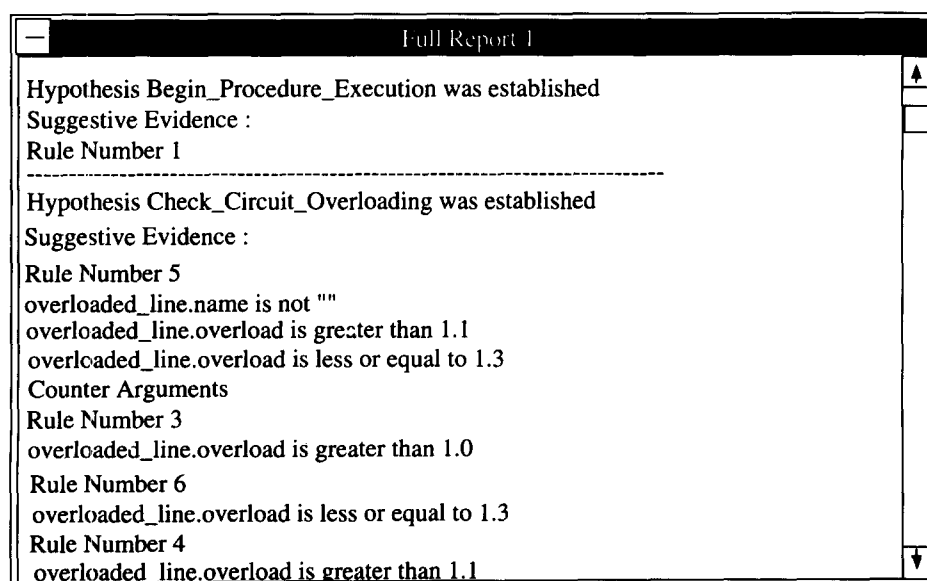


Figure 13. Object network window

expert system shell with graphical representation has also improved the user friendliness of the system and reduced the development period.

For the system to be fully operational, further enhancements and/or modifications may be needed in the light of user experience. The present system is essentially an analysis tool which filters and interprets available information, further research could be undertaken to extend the system allowing it to recommend actions to be taken.

## X. References

- 1 **Sakagushi, T and Matsumoto, K** Development of a knowledge based system for power system restoration' *IEEE Trans. Power Appar. Syst.* Vol PAS-102 No 2 (1983) 320–328
- 2 **Knight, U G** Expert systems in power system planning and operation from the viewpoint of a utility engineer *Proc. of the 9th PSCC Conference*, Cascais, Portugal, 1987, pp. 687–694
- 3 **Suzuki, H, Kawakami, J, Kunugi, M, Tanaka, H and Sekine, Y** 'Experiences of expert systems for power systems analysis in Japan' *Symposium on Expert Systems Application to Power Systems*, Stockholm-Helsinki, August 1988, pp. 1.1–1.7
- 4 **Zitouni, S, Ramsay, B, Moffatt, A and Moyes, I** 'Integration of an expert system with numerical algorithms for power system operational control' *Proc. of the 24th UPEC Conference*, Belfast, September 1989
- 5 **Irving, M R and Sterling, M J H** Efficient Newton–Raphson algorithm for loadflow calculation in transmission and distribution networks *Proc. IEE C* Vol 5 (1987) 325–328
- 6 *Nexpert Object:- User guide, reference manual, database integration guide* Version 2.0, Neuron Data Inc. (January 1991)
- 7 *Microsoft Fortran:- Advanced topics, reference, environment & tools* Version 5.1, MSDOS & OS/2 operating systems (1991)
- 8 *Nexpert Object in the Electric Power Industry* Neuron Data Inc. (1989)
- 9 **Freris, L L and Sasson, A M** 'Investigation of loadflow problem' *Proc. IEE* Vol 115 No 10 (1968)