

# An Intelligent System-based Teaching Tool for Electrical Engineering Subjects

Professor Michael Negnevitsky

*University of Tasmania, Australia, Michael.Negnevitsky@utas.edu.au*

## Abstract

*Computer-based intelligent tutorials are dedicated to support the education of power and electrical engineering students. The aim of this project is to make teaching and learning more productive and efficient by employing modern technologies. It seeks to find new methods to teach large number of students with no increase in staff. Intelligent tutorials are developed based on an expert system shell. They provide the functionally interacted set of theory and problems, and support the student progress monitoring and assessment. This paper describes the development of tutoring software for teaching electrical engineering subjects, and in particular, faults analysis in power systems. The expert system based software has been developed and successfully used by power systems students. They found this software easy to use and understand, and it will become an extra teaching tool.*

**Keywords:** *Intelligent system, Teaching tool, Electrical engineering.*

## 1. Introduction

Engineering education in Australia has gone through a period of rapid changes. [1] In order to create a more technologically competitive society attention now must be paid to quality in higher education. However, Australia has an "engineering intensity" half that of the U.S.A. and 60% that of Japan. Therefore the potential need for engineers is much greater than the present supply.

Rapid growth of quality and quantity demands makes computer-based education crucial. During the last few years computer-based education has been improved by introducing of intelligent tutoring systems. Such systems can be based on Artificial Intelligence (AI) technology. Although some intelligent tutoring systems are already in the stage of practical use, their development in Australia is just in the very initial phase.

An expert system as the base for the intelligent tutoring system provides built-in support facilities such as debugging aids and knowledge base editors, built-in input/output and explanation mechanism. These facilities offer to make the tutoring system development much easier and faster.

The proposed teaching tool operates in an active dialogue mode with the student, uses examples, provides immediate explanations and feedback to students. It also allows the students to access to the learning facility at any time that is convenient for them.

## 2. Computer-based Education

Current tertiary teaching is predominantly conducted through lectures, tutorials, laboratory sessions and workshops. However lectures are not suitable for individualised teaching. Lecturers usually target average students and cannot provide adequate attention to weak and bright ones. Tutorials and laboratory sessions used to be an opportunity for individualised approach. However the steadily increasing ratios of student to staff and a reduction in laboratory work make it difficult to provide reasonable feedback to students on their performance and understanding of the subject materials. Meanwhile, obtaining feedback is the major importance for both teaching and learning. In order to make feedback effective it must be provided promptly and include sufficient explanations.

Computer-based education is able to solve most of these problems. [2] It can introduce the wider range of materials, assess knowledge of a student and, if it is not sufficient, present additional information, provide prompt feedback to the student with explanations of his/her mistakes and, if it is necessary, change the learning format, provide feedback to the lecturer on the material students are having difficulties with and, of course, allow the students access to the computer-based learning at any time. Modern computer-based systems can also allow to incorporate sophisticated multimedia representation of learning materials.

It is a well-known fact that students can learn only by doing or experiencing. There is an old Chinese proverb that to be an invaluable guide for any learning procedure:

"I hear and I forget,  
I see and I remember,  
I do and I understand".

Simply applying theory is ineffective [2] [3]. It is necessary to point out that examples can influence learning process much more than the presentation of concepts and even rules [4].

In order to be successful computer-based tutorials should:

- Determine the prior knowledge of a student;
- Gain and maintain attention and reinforce the motivational state of the student;
- Present examples with clear explanations of key points;
- Provide interactive mode of learning with prompt response to any student answer;
- Provide sufficient and clear explanations of the student mistakes made during tutorial session;
- Provide assessment based on the results achieved and time spent;
- Identify the gap in the student knowledge and, if necessary, present additional materials;
- Provide assessment information to the lecturer.

These principles are used for the tutoring software development.

### 3. Fault Analysis in Power Systems

Fault analysis is one of the basic elements of power system analysis and operation and included in the Power System undergraduate course in electrical engineering. However, a lecturer can spend very few hours to present the calculations of short-circuit currents. For example, at the University of Tasmania, a lecturer should allocate for balanced fault analysis one lecture and one tutorial only. Therefore, some challenging problems cannot be considered.

In order to investigate the scope for the application of a computer-based teaching, a symmetrical fault analysis was accepted.

Fault current in a power system can be determined using fractions of reference quantities, such as rated or full-load values [5]. These fractions are called per unit (p.u.) and used instead of actual values of the quantities. The p.u. value of any quantity can be expressed as [5]:

$$\frac{\text{the actual value (in any unit)}}{\text{the base value (in the same unit)}}$$

Fault current in a simple power system can be calculated in the following order:

1. Choose the appropriate voltage (kV) and complex power (MVA) bases. The choice of the voltage base,  $V_{base}$ , is determined by the voltage rating of the busbar where the fault is located. The power base,  $S_{base}$ , may be taken equal to the power rating of any power system generator or transformer, or can be chosen as a whole number such as 10, 100 or 1000 MVA.

- Calculate the current base (kA)

$$I_{base} = \frac{S_{base}}{\sqrt{3}V_{base}} \quad (1)$$

- Calculate p.u. reactances for generators, transformers, lines, cables and other apparatus. Resistances of these elements usually may be neglected.

For generators

$$x_g = x'' \frac{S_{base}}{S_g} \text{ p.u.} \quad (2)$$

where  $x''$  is the sub-transient reactance in  $\Omega$  and  $S_g$  is the generator rating in MVA.

For transformers

$$x_t = \frac{x\%}{100} \frac{S_{base}}{S_t} \text{ p.u.} \quad (3)$$

where  $S_t$  is the rated quantity in MVA.

For power transmission lines and cables

$$x_l = x_o \cdot L \cdot \frac{S_{base}}{V_{base}^2} \text{ p.u.} \quad (4)$$

where  $x_o$  is the reactance value per phase per km,  $L$  is the length of the line or cable in km and  $V_{base}$  is the voltage base of the line/cable in kV. In fact, it is common to use the rated voltage of the transmission line or cable as  $V_{base}$ .

For current limiting reactors

$$x_r = \frac{V_r}{I_r} \cdot \frac{S_{base}}{V_{base}^2} \text{ p.u.} \quad (5)$$

where  $V_r$  is the voltage drop in V and  $I_r$  is the rated current in A referring to the voltage base of the reactor  $V_{base}$ .

- Calculate the total reactance of the equivalent circuit. The equivalent circuit is presented as a series and series-parallel combinations of the p.u. reactances. This is replaced by the final single equivalent reactance,  $x_{eq}$ . It may also involve the use of the delta-star or star-delta transformations.
- Determine the p.u. value of the short-circuit current

$$I_f = \frac{E}{x_{eq}} \text{ p.u.} \quad (6)$$

where  $E$  is assumed to be 1 p.u.

- Calculate the actual value of the fault current (kV),

$$I_f = I_{fp.u.} \cdot I_{base}$$

(7)

7. Finally obtain the short-circuit level (MVA)

$$S_f = \frac{S_{base}}{x_{eq}}$$

(8)

where  $x_{eq}$  is the reactance from the source of the voltage to the point of the fault in p.u.

Therefore, short-circuit current calculations in power systems involve several steps which must be undertaken by the students.

#### 4. Leonardo Expert System Shell

*Leonardo* expert system shell has been chosen among several packages for development of the prototype application. *Leonardo* is an object oriented tool for developing expert system applications. The *Leonardo* Development System provides the facilities for building and testing expert systems, the *Leonardo* Run System allows to deliver complete applications to the user and the *Leonardo* Productivity Toolkit provides efficient aids for tailoring the application. Figure 1 shows the basic parts of the *Leonardo* expert system shell.

The knowledge base comprises the rules and objects which *Leonardo* uses for representation of the expertise. The knowledge base can store objects with frames which hold additional information, for example, procedures and screen design.

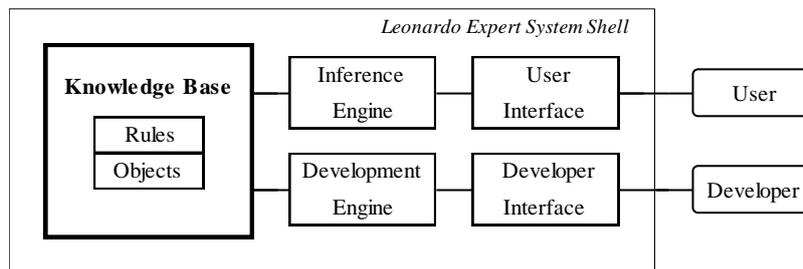


Figure 1. Basic components of the *Leonardo* expert system shell.

The inference engine provides a production rule interpreter, method for finding values of objects, and How? and Why? explanation facilities.

The development engine gives the facilities necessary to create and edit rules for the knowledge base and to edit information into object frames.

There are three types of rules: normal, assertive and quantified. A normal rule tests the values of some objects in its antecedent and sets the values of some objects in the consequent. An assertive rule is a simple command which defines the goal object (*seek*), collects initial data (*ask*) or prints information (*say*). A quantified rule can generate an exhaustive search over all members of the class objects. Class objects may be created with several member objects inheriting certain slots. Multiple inheritance is also permitted for class objects.

*Leonardo* automatically generates screens for the user input and output. However, a screen design package is also available to create tailored input and output screens. The graphics package provides built-in procedures to write a wide range of graphic images to the screen.

*Leonardo* has its own procedural programming language which is used to perform complex computations, access external databases and programs developed in C, FORTRAN and Pascal, print reports, manipulate the screen and so on. Besides *Leonardo* provides a mechanism for uncertainty management.

All these features make *Leonardo* a very inviting shell for development of expert systems, especially in the stages of idea and prototype.

## 5. Design of the Tutoring System

*Leonardo* expert system shell has been chosen among several packages for development of the prototype application. *Leonardo* is an object oriented tool for developing expert system applications. The *Leonardo* Development System provides the facilities for building and testing expert systems, the *Leonardo* Run System allows delivering complete applications to the user and the *Leonardo* Productivity Toolkit provides efficient aids for tailoring the application. Figure 1 shows the basic parts of the *Leonardo* expert system shell.

In order to illustrate details involved in the design of the expert system, let us consider a tutorial on fault analysis in power systems.

The tutorial session starts with the introduction screen which provides a start-up message. This message is placed in the *Introduction* slot of the object targeted by the *seek* command in the main rule set. When execution begins the corresponding text will be found and displayed on the screen as shown in Figure 2.

Then the illustrative example is demonstrated. It provides step-by-step calculations of the symmetrical fault current in the power system. Typical screens presented to the student are shown in Figure 3. The solution is easy to follow and understand.

The student is then asked whether or not he/she wants to run this example again. If the example is understood, the user is taken to the standard query screen shown in Figure 4. The actual text which appears on this screen is determined by the corresponding slots of the object in question. The query preface area on the top half of the screen is used to give the user information about the query. If the *QueryPreface* slot has no text in it, the query preface area of the screen will be blank. The *QueryPrompt* slot contains a text which is displayed on the middle line of the screen. If the *QueryPrompt* slot is empty, a default prompt will be generated.

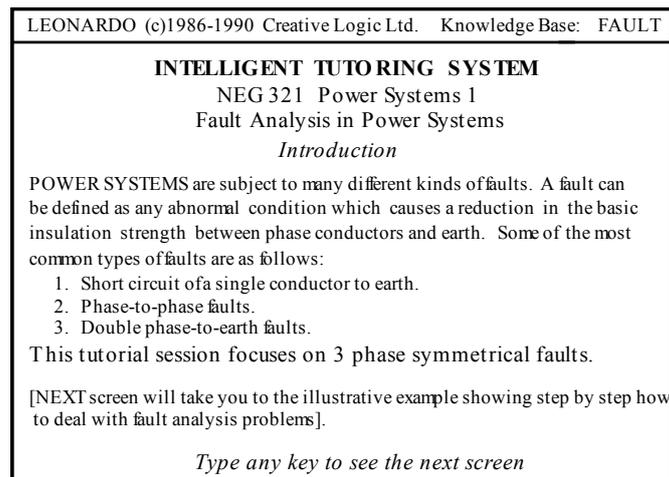


Figure 2. Introduction screen.

### FAULT ANALYSIS EXAMPLE

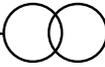
This example illustrates calculations of the fault current ( $I_f$ ) and fault MVA ( $S_f$ ) for the power system shown below.

Generator



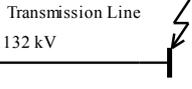
$S_g = 60 \text{ MVA}$   
 $X'' = 0.1$

Transformer



$S_t = 60 \text{ MVA}$   
11/132 kV  
 $X = 8\%$

Transmission Line



132 kV  
Length = 100 km  
 $x_0 = 0.1 \text{ Ohms/km}$

Fault

In order to achieve correct solution the following steps should be undertaken:

*Type any key to see the next screen*

---

**STEP 1:**

- Choose your S base.
- \* The power base can be adopted equal to the power rating of any power system generator or transformer, or may be chosen as a whole number as 10, 100 or 1000 MVA.

**S base is chosen as 60 MVA.**

**STEP 2:**

- Choose your V base.
- \* The choice of the voltage base is determined by the voltage ratio of the busbar where the fault is located.

**V base is chosen as 132 kV.**

*Type any key to see the next screen*

Figure 3. The step-by-step solution screens.

The input area under the prompt line includes the allowed input values for the object and the explanatory text for each individual value. This area is controlled by the *AllowedValue* and *AVExpansion* slots. When the query screen is displayed, the first option on the input menu is highlighted.

The student can select and then solve any of the problem displayed on the query screen. Seven problems were developed. Each of them uses a random number generator to provide different values for the variables. Problem difficulties range from easy (fault 1) to challenging (fault 7).

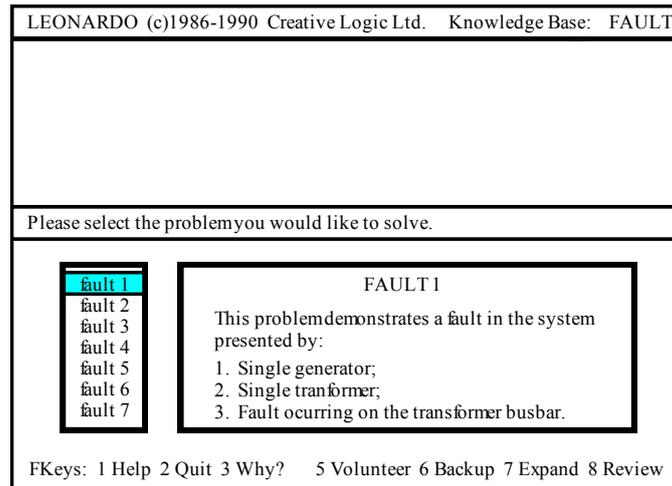


Figure 4. Format of the standard query screen.

## 6. Conclusion

The expert system based tutoring system for teaching electrical engineering subjects has been developed and successfully used by the third year students. They found this software easy to use and understand and received a good introduction to fault analysis in power systems.

## 7. Acknowledgements

The author wish to thank engineering graduates Mr Nicholas Cole and Mr Ambika Sharma for their contribution in the design of the intelligent tutoring system.

## References

- [1] J Webster, "Engineering Education in Australia", *International Journal of Engineering Education*, Vol. 16, No. 2, pp. 146-153, 2000.
- [2] J. Lowe, "Computer based education: Is it a panacea", *Journal of Research on Technology in Education*. Winter 2001/2002, Vol. 34 Issue 2, pp. 163-171.
- [3] G. Gibbs and T. Habeshaw, *Preparing to Teach - An Introduction to Effective Teaching in Higher Education*, Technical and Educational Services Ltd, 1989.
- [4] R. J. Sternberg, *Cognitive Psychology*, 6<sup>th</sup> edition, 2011.
- [5] B. M. Weedy, B. J. Cory, N. Jenkins, J. B. Ekanayake and G. Strbac. *Electric Power Systems*, 5<sup>th</sup> edition, Wiley, 2012.

## Author

**Michael Negnevitsky** received the B.S.E.E. (Hons.) and Ph.D. degrees from the Byelorussian University of Technology, Minsk, Belarus, in 1978 and 1983, respectively. Currently, he is Chair Professor in Power Engineering and Computational Intelligence and Director of the Centre for Renewable Energy and Power Systems at the University of Tasmania, Hobart, Australia. From 1984 to 1991, he was a Senior Research Fellow and Senior Lecturer in the Department of Electrical Engineering, Byelorussian University of Technology. After arriving in Australia, he was with Monash University, Melbourne, Australia. His interests are power system analysis, power quality, and intelligent systems applications in power systems. Dr. Negnevitsky is a Chartered Professional Engineer, Fellow of the Institution of Engineers Australia, and Member of CIGRE AP C4 (System Technical Performance), Member of CIGRE AP C6 (Distribution Systems and Dispersed Generation), Australian Technical Committee, and Member of CIGRE Working Group JWG C1/C2/C6.18 (Coping with Limits for Very High Penetrations of Renewable Energy), International Technical Committee.