*Int. J. Appl. Engng Ed.* Vol. 7, No. 2, pp. 151–157, 1991
Printed in Great Britain.

0742–0269/91 $3.00+0.00


# Applied Engineering Education using Artificial Intelligence Techniques*†

A. MICARELLI‡
F. MUNGO§
F. S. NUCCI§
*Dipartimento di Informatica e Sistemistica, Università degli Studi di Roma 'La Sapienza', Via Salaria*, 113, *I*-00198 *Roma, Italy*‡
*Laboratorio Intelligenza Artificiale, SELENIA—Industrie Elettroniche Associate S.p.A., Via Tiburtina Km.* 12.400, *I*-00131 *Roma, Italy*§

*Artificial Intelligence techniques are expected to make computer-based educational systems more effective. In this paper an important feature of Intelligent Tutoring Systems (ITS), i.e. the automatic problem solving capability in the domain to be taught, also when the problem is proposed by a student, is stressed. This facility in fact allows the difficulties with traditional CAI (Computer Aided Instruction) systems to be overcome. In the paper two examples of the use of such a technique, one in the domain of classical mechanics, another for the analysis of electrical circuits, are presented.*

## 1. INTRODUCTION

INTELLIGENT Tutoring Systems (ITS) are effective tools to support human learning. Distinguishing features of ITSs, compared with traditional Computer Assisted Instruction (CAI) systems, are (i) automatic problem solving capabilities in the domain to be taught, (ii) mixed-initiative interaction in the teaching/learning relationship, (iii) personalization of the teaching interaction to the needs of each student.

The capability of autonomously determining all the possible solutions to problems, also when they are proposed by the student, is an essential feature and winning point as to traditional applications. In fact, it allows the educational system to explain why one solution is better than another or why and where a solution procedure is wrong. In general, this capability can be obtained especially when the domain to be taught is formalized: so the domains of applied engineering are well suited for the realization of educational systems with the previously mentioned characteristics.

In this paper we want to stress the improvement it is possible to obtain for the realization of more effective ITSs in these domains using a particular inferential mechanism, based on *contexts* and *Truth Maintenance System* (TMS).

The paper is organized as follows. In Section 2 we first sketch the general architecture of ITSs. In Section 3 two examples of the use of a contexts mechanism in the domain of engineering, are presented. The first one is relative to mechanics, the second is relative to electrical engineering, and is taken from a real session with an ITS for teaching students how to analyze electrical circuits, developed at the University of Rome 'La Sapienza'. In the concluding Section, we present a few final remarks.

## 2. THE ARCHITECTURE OF INTELLIGENT TUTORING SYSTEMS

Generally speaking, an Intelligent Tutoring System can be viewed as a family of Expert Systems, each of them specialized in a particular aspect of the teacher's activity. This activity requires expertise in the domain to be taught, on the teaching methodologies and on ways to personalize the teaching dialogue to the needs of each student.

In order to fulfil these requirements, the general architecture of an ITS has to contain the following components: an expert module in the subject domain, a tutorial module, a student model and an interaction component (see Fig. 1).

The *expert module* is an expert system in the domain to be taught, capable of determining the correct solution for the problems proposed to the student during the teaching session.

The *tutoring module* encompasses the teaching expertise, intended both as a diagnostic capability in determining the reasons for the student's errors, and as a capability in evaluating the learning status of the student and choosing the teaching strategy which seems to be the most suited to achieve the present goals.

* This work has been partially supported by CUD (Consorzio per l'Università a Distanza).
† Paper accepted 9 October 1990.

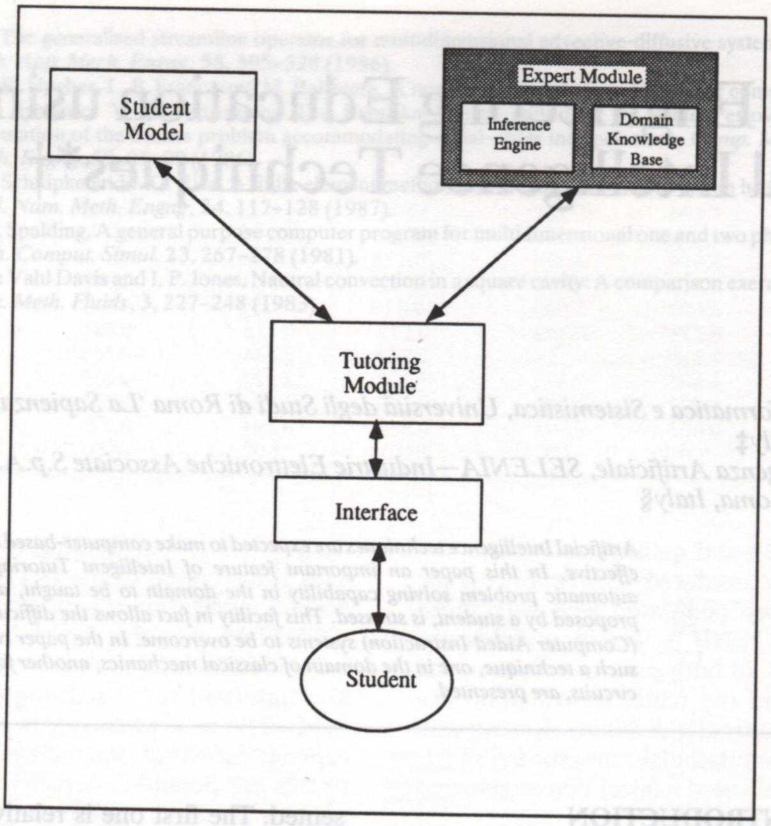


Fig. 1. General architecture for ITSs.

The *student model* is a knowledge base concerning the learning status of the student as gathered by the system through the dialogue. It is a *qualitative model*, dynamically improved and increased, where the knowledge acquired by the student is represented explicitly. This module allows the system to personalize the teaching dialogue: in fact, the best teaching action is chosen by the tutoring module on the basis of the model of the student presently using the system.

The *interaction module* plays an important role in determining the effectiveness of the educational system: it must make the system easy to use, provide a pleasant interaction, but, above all, it must allow the student to take the initiative in asking questions, requesting explanations and suggestions, thus realizing a *mixed initiative* dialogue [1, 2, 3, 4].

## 3. ITSs IN ENGINEERING EDUCATION

As an example of the potential offered by Artificial Intelligence techniques for educational purposes in the domain of engineering, we present two expert modules: the first is relevant to the solution of problems about dynamics of material points; the second is relevant to the analysis of electrical circuits with alternating current in steady condition.

Both of them have been implemented in KEE (Knowledge Engineering Environment), a sophisticated environment for the development of knowledge based systems, running in LISP on an EXPLORER workstation. In the implementations, a *contexts mechanism* associated with a *Truth Maintenance System* has been used.

Before entering into the details of the examples, we think is right to present briefly the inferential mechanism used.

In a context mechanism, the basic structure provided for modeling actions is a direct acyclic graph of worlds. A world represents a set of related facts—for example a situation, a belief set, an action or a hypothetical state of a problem solver. The problem solver is constantly confronted with the need to select among equally plausible alternatives. These may concern the selection of which goal to try to achieve next, which formulae to use in a hypothetical reasoning, etc. Often most of these alternatives will be incorrect. In this view, the problem solver is searching a space where each dimension is defined by the set of alternatives it has encountered. Then, the problem is the efficiency in searching the space of alternatives. A world, together with its ancestors in the graph, represents a partially ordered network of actions. Each successor of a world in a graph then represents a hypothetical extension of the world's associated action network to include a new subsequent action. Problem solving is an exploratory incremental process. That is, one starts with a set of beliefs, that are true in every situation and that represent the static knowledge base (in KEE this Knowledge

Base is called Background), and recursively considers alternative choices that modify those beliefs. Usually, the modifications to a set of beliefs are incremental. The TMS detects the inconsistent worlds, i.e. the worlds where there is a violation of some constraints or where two facts are in contradiction. When a world becomes inconsistent it cannot give more information and cannot generate more children worlds. This makes it possible a very effective search in the search space.

Another advantage derived from the use of the world system is the possibility to create *world merges*. A merge world is a world that has more than one parent. The ability to perform merges allow a problem to be decomposed into nearly independent components, which can be worked separately and later recombined [5].

3.1 *An example in the domain of mechanics*

In this problem, presented in Fig. 2, the initial data are: the starting speed $V_0$, the material point mass $m$, the angle of the inclined plane $\alpha$.

In the expert module, the declarative knowledge relevant to the objects present in the domain is represented using the formalism of the *frames*. A frame is a data structure used to describe a stereotyped object, act or event in terms of their properties (or attributes): for instance, the attributes of a material point are the mass, the initial position, the initial speed, etc.

The deductive knowledge is represented as a taxonomy of production rules which implement the knowledge relative to the physical laws of mechanics. A production rule has the following form:

IF ⟨conditions⟩ THEN ⟨conclusions⟩.

The execution of a production rule means the application of the corresponding physical law to a given problem.

The goal of the exercise in Fig. 2 is the determination of the space l covered by the material point when it inverts the motion (i.e., when the ending speed is: V = 0).

The expert module uses the initial data and executes in *forward chaining* the production rules, representing the applicable physical laws, to determine the solutions to the problem.

In Fig. 3 the complete graph (built by the system) of all the possible correct paths to the solution is shown. Each node, that corresponds to a different World, represents the application of a different physical law. For example, the World n. 1 represents the application of the formula for the computation of the potential energy of the point $P$ at the starting time ($U_0 = mgh_0$), and is produced by the following rule:

*(IF*
*(?P IS IN CLASS POINT)*
*(THE MASS OF ?P IS ?M)*

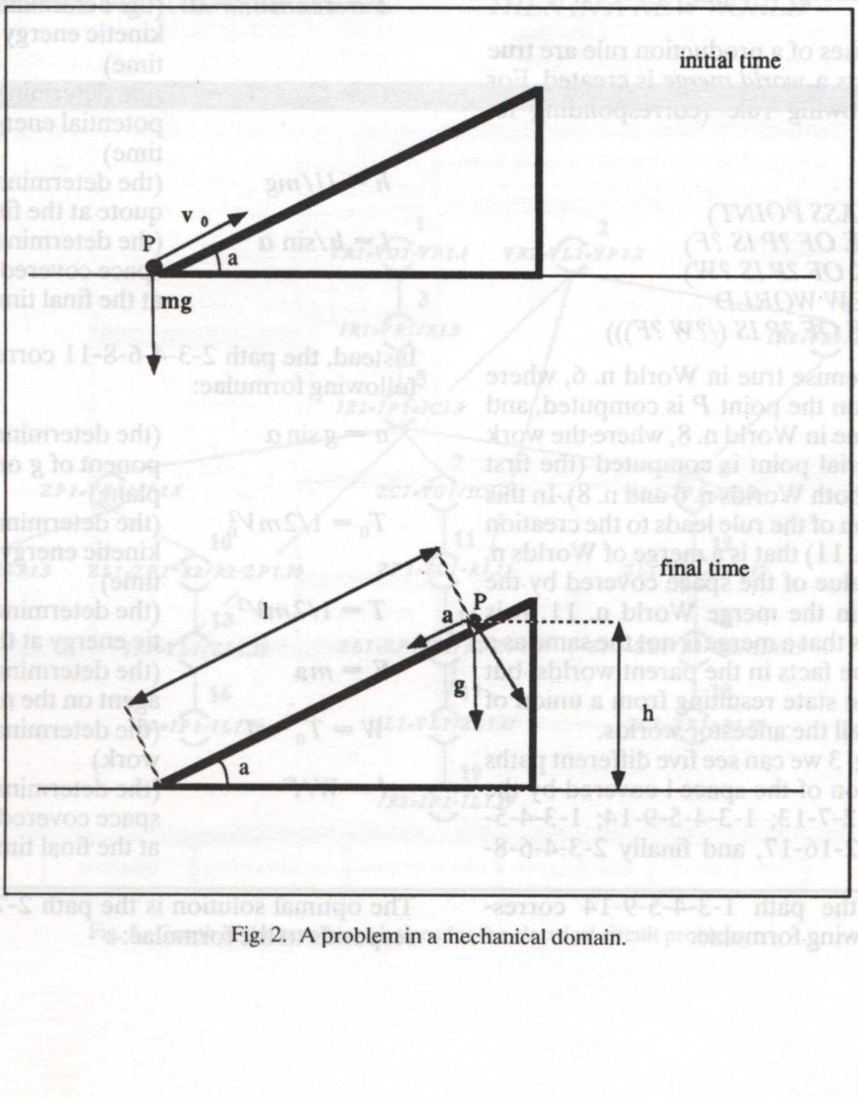


Fig. 2. A problem in a mechanical domain.

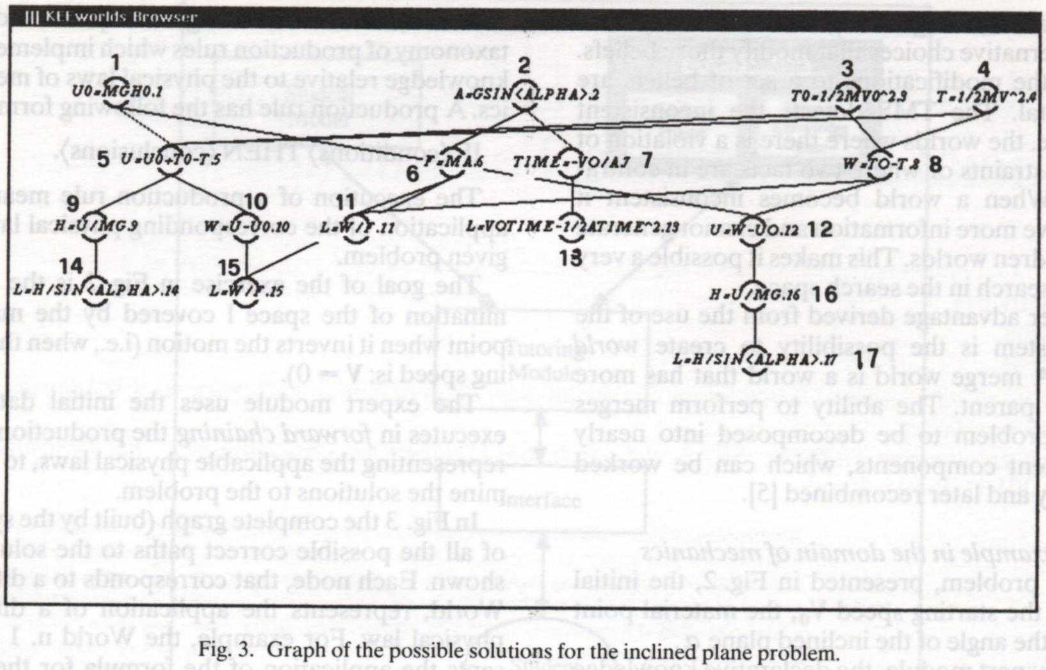


Fig. 3. Graph of the possible solutions for the inclined plane problem.

(*THE QUOTE OF ?P IS ?H*)
*THEN IN A NEW WORLD*
(*THE POTENTIAL ENERGY OF ?P IS* (**?M ?H 9.81*)).

In this way in the World n. 1 is asserted that the starting potential energy is equal to zero (because $h_0 = 0$).

When the premises of a production rule are true in different contexts a *world merge* is created. For example, the following rule (corresponding to: l = *W/F*):

(*IF*
(*?P IS IN CLASS POINT*)
(*THE FORCE OF ?P IS ?F*)
(*THE WORK OF ?P IS ?W*)
*THEN IN A NEW WORLD*
(*THE SPACE OF ?P IS* (*/?W ?F*)))

has the second premise true in World n. 6, where the force applied on the point *P* is computed, and the last premise true in World n. 8, where the work done by the material point is computed (the first premise is true in both Worlds n. 6 and n. 8). In this case the application of the rule leads to the creation of a new World (n. 11) that is a merge of Worlds n. 6 and n. 8. The value of the space covered by the point is asserted in the merge World n. 11. It is important to stress that a merge is not the same as a simple union of the facts in the parent worlds, but rather produce the state resulting from a union of the changes from all the ancestor worlds.

If we look at Fig. 3 we can see five different paths for the computation of the space l covered by the point. They are: 2-7-13; 1-3-4-5-9-14; 1-3-4-5-10-15; 1-3-4-8-12-16-17, and finally 2-3-4-6-8-11.

For example, the path 1-3-4-5-9-14 corresponds to the following formulae:

$U_0 = mgh_0$ (the determination of the potential energy at the starting time)

$T_0 = 1/2mV_0^2$ (the determination of the kinetic energy at the starting time)

$T = 1/2mV^2 = 0$ (the determination of the kinetic energy at the final time)

$U = U_0 + T_0 - T$ (the determination of the potential energy at the final time)

$h = U/mg$ (the determination of the quote at the final time)

$l = h/\sin\alpha$ (the determination of the space covered by the point at the final time).

Instead, the path 2-3-4-6-8-11 corresponds to the following formulae:

$a = g\sin\alpha$ (the determination of component of $g$ on the inclined plane)

$T_0 = 1/2mV_0^2$ (the determination of the kinetic energy at the starting time)

$T = 1/2mV^2$ (the determination of kinetic energy at the final time)

$\mathbf{F} = m\mathbf{a}$ (the determination of force agent on the material point)

$W = T_0 - T$ (the determination of the work)

$l = W/F$ (the determination of the space covered by the point at the final time).

The optimal solution is the path 2-7-13, that corresponds to the formulae:

$a = g \sin\alpha$ (the determination of component of $g$ on the inclined plane)
$t = V_0/a$ (the determination of the final time)
$l = V_0 t - 1/2at^2$ (the determination of the space covered by the point at the final time)

We wish to remark on the fact that the expert module is capable of determining all the solutions to a problem, also if the problem is proposed by the student. This is due to the use of an explicit representation of the knowledge of the domain that, associated with an inferential mechanism, endows the expert module with problem solving capabilities.

### 3.2 *An example in the domain of electrical engineering*

Also for this example, the declarative knowledge pertinent to the topology and the electrical components of the circuit to be analyzed is represented via a hierarchy of frames. The deductive knowledge is represented as a taxonomy of production rules that implement the knowledge relevant to the applicable electrical laws and formulae, such as Ohm's law, Kirchhoff's laws, equivalent impedance formulae, etc. The system uses the *symbolic method* (electrical quantities represented as complex numbers) in its problem solving activity.

The circuit is shown in Fig. 4: the initial data are the voltage of the generator $\mathbf{V}_G$, the value of the resistors $R_1$ and $R_2$, and the voltage of the shunt $V_{P1}$. The goal of the problem is the determination of the value of the equivalent impedance of the whole circuit. The expert module uses initial data and the electrical laws, represented by the production rules, to obtain the complete solution graph. In Fig. 5 the worlds structure, representing the deductive logical paths to all the unknown data, is shown. Every world corresponds to the use of a physical law and is created by the execution of a production rule. For example, the World n. 2 represents the application of the second Kirchhoff law and is produced by the following rule:

*(IF*
*(?S IS IN CLASS SHUNT)*
*(THE VOLTAGE OF ?E IS ?V)*
*THEN IN A NEW WORLD*

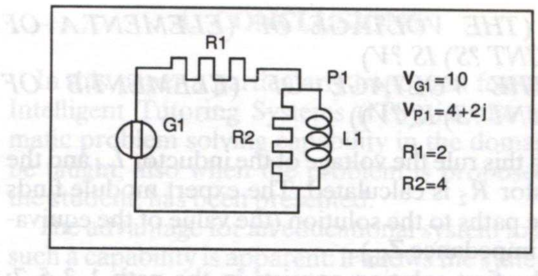

Fig. 4. Electrical circuit problem.

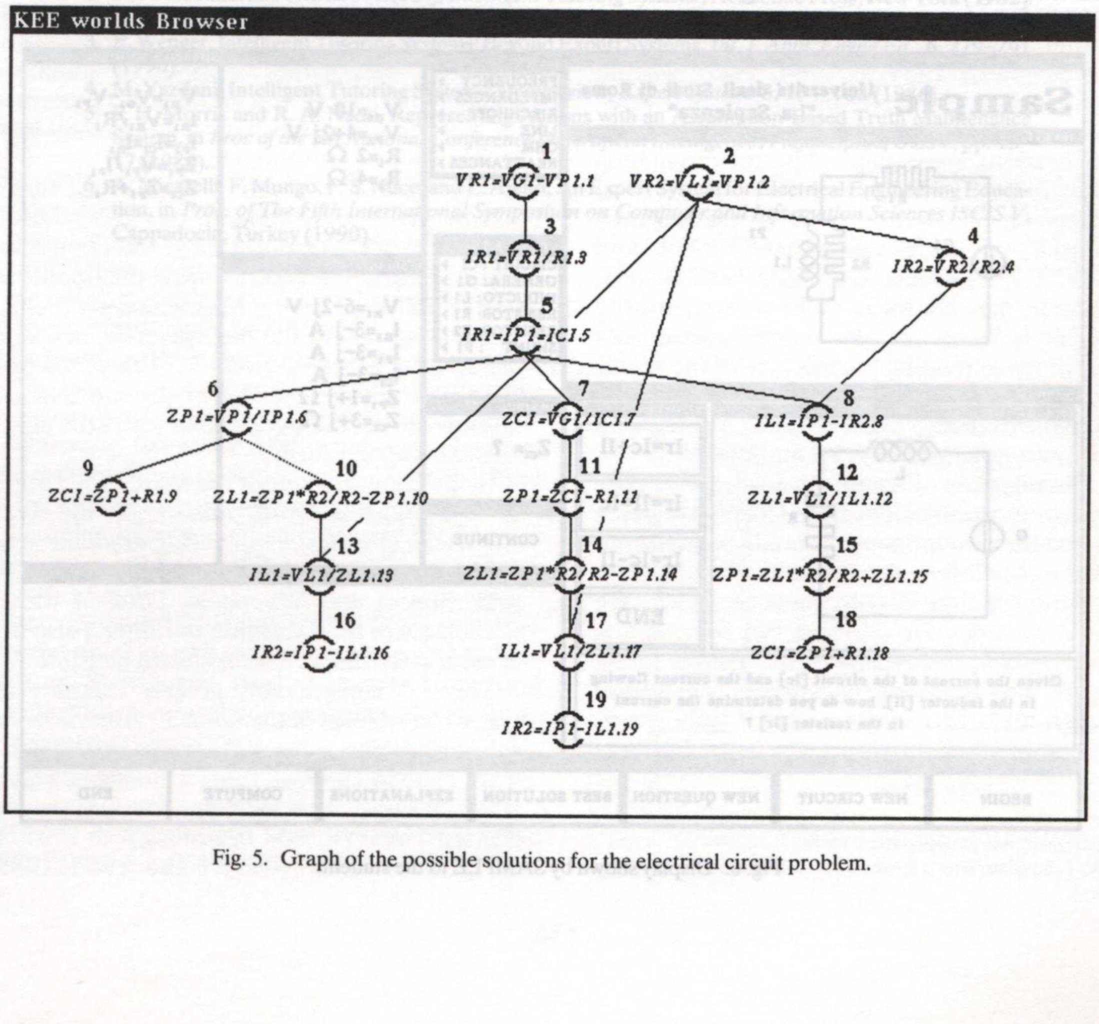

Fig. 5. Graph of the possible solutions for the electrical circuit problem.

(*THE VOLTAGE OF* (*ELEMENT.A OF SHUNT ?S*) *IS ?V*)
(*THE VOLTAGE OF* (*ELEMENT.B OF SHUNT ?S*) *IS ?V*))

With this rule the voltage of the inductor $L_1$ and the resistor $R_2$ is calculated. The expert module finds three paths to the solution (the value of the equivalent impedance $\mathbf{Z}_{eq}$).

The first solution consists in the path 1-3-5-7; this solution is the shortest one and corresponds to the following formulae:

$\mathbf{V}_{R1} = \mathbf{V}_{G1} - \mathbf{V}_{P1}$ (second Kirchhoff's law)
$\mathbf{I}_{R1} = \mathbf{V}_{R1}/R_1$ (Ohm's law)
$\mathbf{I}_{C1} = \mathbf{I}_{R1} = \mathbf{I}_{P1}$ (first Kirchhoff's law)
$\mathbf{Z}_{eq} = \mathbf{V}_{G1}/\mathbf{I}_{C1}$ (Ohm's-generalized law)

The second solution consists in the path 1-3-5-6-9; finally, the third solution in the path 1-2-3-4-5-8-12-15-18.

This example has been taken from the expert module of SAMPLE [6], an ITS for teaching high school students how to analyse alternating current circuits in steady condition.

Naturally the expert module is only a chunk of the whole system. In order to investigate how the expert module interacts with the other modules, we now present briefly the architecture of the system.

Living out the expert module, SAMPLE is constituted by:

— A *student model*, which contains a representation of the learning status of the student.
— A *tutoring module*, consisting of a *control subsystem*, which monitors the interaction with the student, and by a *remedial subsystem*, which attempts to tailor the right remediation to the needs of the student.
— A *bug catalogue*, which is a library of common errors made by the students.
— A *user interface*.

At the beginning of a teaching session the system either proposes a circuit to be analyzed or waits for a circuit proposed by the student. At this point the expert module determines the correct solution(s).

The system then gives the initiative to the student. In Fig. 6 the display shown by SAMPLE to the student during this part of a teaching session is presented. The actions performed by the student are monitored step by step by the tutoring module and compared with the solution(s) built by the expert module. If the student makes a mistake or is not able to continue, the tutoring module prompts a suitable correction or a hint or an *ad hoc* example (according to the content of the student model) and guides the student back to the correct path. At the same time the student model is updated. During the teaching session, the student is allowed to reason both forwards and backwards, and his reasoning process is presented in the corresponding window.

The system monitors the student step by step comparing his resolution process of the exercise with the solution derived by the expert module. At each step, it analyzes the action performed by the student and possibly decides the corresponding teaching actions, in case the student needs to be

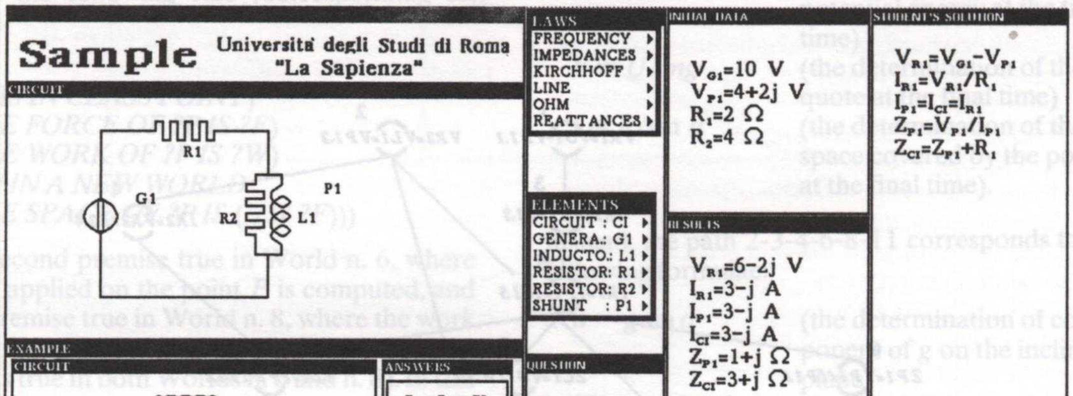


Fig. 6. Display shown by SAMPLE to the student.

guided back to the correct path. If the student cannot find the correct next step to the solution, the remedial subsystem tries to stimulate him in doing reasoning by analogy.

The remedial subsystem performs its goal by singling out the situation that brought the student to make the error and searching from the student model data about precedent behaviour of the student in analogous situations. According to this information, the remedial subsystem decides either to give a refresher about the theory, or to ask a simple question about an example that simplifies the situation currently under focus. In this way it is possible to give suggestions to students to increase their ability in decomposing a problem into sub-problems, and in the search for the analogy.

SAMPLE is a research prototype, and has been tested in experimental settings with school students, who seemed to enjoy the interaction and the use of complex numbers as a representation method for the analysis of electrical circuits. We noticed that the use of the system improved especially the quickness in reaching a satisfactory knowledge in the relevant domain. Such tests have also played a fundamental role for the collection and the analysis of student misconceptions, which presently constitute the bug catalogue of the system.

## 4. CONCLUSIONS

In this paper a particularly important feature of Intelligent Tutoring Systems (ITS), i.e. the automatic problem solving capability in the domain to be taught, also when the problem is proposed by the student, has been presented.

The advantage for an educational system to have such a capability is apparent: it allows the system to explain why one solution is better than another, why and where a solution is wrong, and allows the student to take the initiative in asking questions, requesting explanations, proposing new problems, thus realizing a mixed initiative dialogue.

The domains of applied engineering are well suited for the realization of tutoring systems with these features.

In the paper we have presented two examples of expert modules for such systems, one in the domain of mechanics, one in the domain of electrical engineering. For the realization of these modules a powerful inferential mechanism, based on contexts and Truth Maintenance System, has been used.

To conclude, we can say that the presented approach for engineering education offers great benefits that would complement traditional Computer Aided Instruction (CAI) systems.

## REFERENCES

1. D. H. Sleeman and J. S. Brown (eds), *Intelligent Tutoring Systems*, Academic Press, New York (1982).
2. E. Wenger, *Artificial Intelligence and Tutoring Systems*, Morgan Kaufmann (1987).
3. E. Wenger, Intelligent Tutoring Systems: Beyond Expert Systems, *Int. J. Appl. Engng Ed.*, **6**, 279–291 (1990).
4. M. Yazdani, Intelligent Tutoring Systems: an overview, *Expert Systems* **3**, 154–162 (1986).
5. P. H. Morris and R. A. Nado, Representing Actions with an Assumption-Based Truth Maintenance System, in *Proc of the 5th National Conference on Artificial Intelligence*, Philadelphia, U.S.A., pp. 13–17 (1986).
6. A. Micarelli, F. Mungo, F. S. Nucci and L. Aiello, An Expert System for Electrical Engineering Education, in *Proc. of The Fifth International Symposium on Computer and Information Sciences ISCIS V*, Cappadocia, Turkey (1990).