# Intelligent distance diagnosis of students' solutions. DIRCE: Diagnostic Interactive Relationship-Causality Engine.

Emmanuel Barbounis, Maria Grigoriadou and George Philokyprou

Department of Informatics, University of Athens, Athens, Greece

One of the important and most difficult tasks of open distance learning is the efficient diagnosis of students' solutions of given problems and exercises. Intelligent diagnostic systems and tools may essentially assist in this type of diagnosis by employing new methods and techniques from the fields of Artificial Intelligence and Cognitive Psychology. In this paper we present DIRCE (Diagnostic Interactive Relationship-Causality Engine), a novel intelligent diagnostic engine suitable for computer-based distance diagnosis. The scope of DIRCE is the detection of discrepancies in the students' solutions, and the identification of relevant errors. DIRCE diagnoses problems solved in a procedural manner with steps interconnected through mathematical equations. The engine does not need extended student models or bug libraries. It is mainly based on the correct solution(s) of the problem, the interconnections among the steps of the student's solution, as well as on limited evidential information about the solution under examination acquired interactively from the user. DIRCE has been used in the DIAS and I-DIAS experimental systems diagnosing the students' solutions of problems in high school physics and chemistry.

*Keywords:* Distance Learning, Remote Diagnosis, ITS

## 1. Introduction

The development of open distance learning is one of the major challenges of the coming decades. This form of education and training combines the interaction between the student and teacher ordinary face-to-face instruction with large components of independent or autonomous learning. Distance teaching and group training by means of interactive simulations and virtual learning environments, video-conferencing, computer-assisted conference networks and multimedia communications activate an educational reform (Hunter, 1993).

A very important field of this type of education is the distance diagnosis of students' solutions. It helps a tutor, human or artificial, to understand a learner's level of knowledge by remotely examining his/her performance (responses, problem solutions, etc.). The development of distance diagnosis has to cope with quite a number of intrinsic particularities and limitations. For example, a tutor can instruct hundreds of learners in distance teaching but in no case he can diagnose their errors and suggest solutions for each one individually. Intelligent diagnostic systems and tools may essentially assist this type of diagnosis by employing new methods and techniques from the domains of Artificial Intelligence and Cognitive Psychology. The requirements and constraints of computer-based distance diagnosis are implied by considering the limitations and prerequisites of distance learning (Foster, 1992; Hunter, 1993) and computer-based intelligent diagnosis (Ohlsson, 1993; VanLehn, 1988). Among the most significant of them are:

- *Reliability and speed.* Dominating quality of any "artificial diagnostician" is the quick and correct diagnosis of the students' errors.
- *Interactive diagnosis of the learner's solution.* Students usually solve the given problems and exercises in their environments by using their own, familiar tools. Therefore, an artificial di-

agnostician must perceive learners' solutions by interactively acquiring the required information.

- *Extended knowledge domains.* Distance diagnosis concerns a great variety of topics and knowledge domains that are not usually diagnosed by static and monolithic diagnostic techniques and methods.
- *Laborious knowledge acquisitions.* Human expertise acquisition, especially that of the diagnosticians, is a very difficult process (Keyes, 1989), either because experts are not able to properly and efficiently express their expertise to the knowledge engineers, or due to severe problems in the transformation of the human way of thinking to computer operation.
- *Multiple learners concurrent diagnosis.* Students are working concurrently in an educational network. Therefore, a diagnostic system must be able to diagnose efficiently a learner's responses and solutions, independently of the active diagnostic sessions of other learners.
- *System resources.* Among the major problems of on-line and real-time systems are:
  - the availability of resources (knowledge bases, programs, etc.),
  - the distribution of these resources to the users.

In this paper DIRCE (Diagnostic Interactive Relationship-Causality Engine), an intelligent diagnostic engine applicable to computer-based distance diagnosis of students' solutions is presented. DIRCE deals with the problems solved in a procedural manner, with steps interconnected through mathematical equations. This engine has been used in I-DIAS (Intelligent-Diagnostic Instruction Assistant System) a diagnostic system under evaluation (Barbounis & Philokyprou, 1995), that examines and diagnoses students' problem solutions in high school physics and chemistry.

## 2. Intelligent diagnostic approaches

A variety of methods and techniques has been developed for the intelligent diagnosis of students' and trainees' responses and solutions. The great majority of these techniques are categorised under three main approaches:

**A. Overlay modelling.** These methods and techniques consider the student model as a proper subset of the expert model. For a given problem, they presuppose the existence of the expert's solution that is compared to the student's solution. The detected discontinuities and differences identify the missing conceptions of the student but not the misconceptions. The overlay modelling techniques have a relatively easy implementation however, they can fail when the student's solution, correct or erroneous, differs greatly from the expert's solution. They have been applied in a number of educational systems like the GUIDON (Clancey, 1982) and WEST (Burton & Brown, 1982).

**B. Bug library and machine learning approaches.** The bug library approach diagnoses a student's solution by employing a library containing a set of bugs related with the problem under examination. The diagnostic module compares the behaviour predicted by the bugs, with the discrepancies observed in the student's solution. The resulting diagnosis corresponds to the identification of a subset of the bug library able to explain all the existing errors in the specific solution. The biggest obstacle of the approach is the building up of the bug library. If the library is not sufficient, the solution under examination may be totally misdiagnosed. Furthermore, an empirical study by Payne and Squibb (1990) indicates that bugs can vary across the student populations. This may result in the need to reconstruct bug libraries for each new student population, a fact that is quite impractical. The machine learning techniques have been developed to avoid the painful process of constructing bug libraries (Ohlsson & Langley, 1988; Langley et al, 1990). This approach infers bugs bottom-up during the diagnostic operation. It is usually assisted by auxiliary libraries of elements that may be combined for the construction of the bugs (VanLehn, 1988). Diagnosis based on both approaches is computationally expensive.

**C. Model tracing.** Model tracing is a practical and easily realisable diagnostic technique. According to this approach, a student proceeds from a mental state to the next one by firing

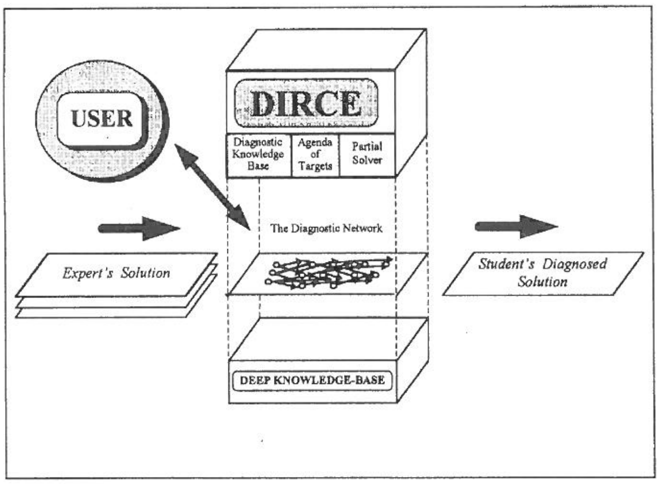


*Fig. 1.* The operating environment of DIRCE

some rule, correct or false. Furthermore, it assumes that all the student's mental states in a solution under examination are accessible and that the corresponding rules are available to the diagnostic module. The technique traces the student's solution by problem solving and diagnosing each individual step. It identifies the rule evoked by the student in the particular step based on a corresponding rule library. Model tracing is used by a number of intelligent tutoring systems (Anderson, 1988; Anderson et al., 1990). The obligation to diagnose each faulty step as it occurs limits the usefulness of the approach from the pedagogical side of view (Ohlsson, 1993), while the construction of sufficient rule libraries may request thorough empirical research. Moreover, the necessity to attend and coach the student during the solution process restricts the applicability of the technique in distance diagnosis.

## 3. Overview of DIRCE

The goal of this engine is the identification of all the errors in the students' solutions, with speed and effectiveness. Its major characteristic is the interactive identification of faults, by acquiring necessary information from the users in a minimum but adequate number of diagnostic steps. Prerequisites for the application of DIRCE as shown in Fig. 1 are the following components:

- *The correct solutions data base.* It comprises the experts' solutions for the given problems. It represents the shallow knowledge of the system. During the diagnostic process, a properly selected correct solution is activated and used for the diagnosis of the student's solution. Fig. 2 shows the record structure and the description of a step in the data base of correct solutions.

| | |
|---|---|
| Step name: | Distance covered by a car $A$ |
| Meaning: | Distance |
| Expert symbol: | $x_A$ |
| Correct relation: | $\frac{1}{2}a_{A1}t_{A1}^2 + u_{A1}t_{A2}$ |
| Correct components | $[a_{A1}, t_{A1}, u_{A1}, t_{A2}]$ |
| Correct value: | 217.8 |
| Measurement unit: | km |
| Error Causing Belief: | 0.8 |

*Fig. 2.* Record structure of a step

- *The diagnostic network.* When a diagnostic process commences, the engine utilising the chosen correct solution of the problem under examination constructs a diagnostic network. The specific procedural acyclic network constitutes the heart of the diagnostic operation and its structure is similar to the structure of a Bayesian network (Pearl, 1986; Keung-Chi & Abramson, 1990; Peot & Shachter, 1991). Each node of the network corresponds to a step of the solution and is depicted by a frame. The erroneous steps are called *fault nodes*. The slots of the frame may contain functions and equations, results, causal factors and coefficients, solution explanations, current diagnostic information, etc. The links that connect a node with its parents are defined by one and only one rule. The output of every node corresponds to a set of values that are transmitted to the direct descendants of the node. Each set of values may be empty or it can contain one or more elements. This scheme has been selected to represent the structure and behaviour of the solution under examination, taking advantage of the computational architecture of belief networks for the propagation of pieces of evidence and local diagnostic conclusions.

The network is dynamically transformed as new pieces of evidence are collected during the diagnostic operation. New steps may be added in, existing steps may be altered or clustered into a single step or decoupled into partial steps or eliminated completely, while existing links may be substituted by new ones or removed. Thus, by the end of the process the network corresponds to the student's solution.

Fig. 3 shows the frame structure of the altered step of Fig. 2. The user components are connecting links with the parent nodes, while the user relation corresponds to the interconnecting rule.

| | |
|---|---|
| Step name: | Distance covered by a car $A$ |
| Meaning: | Distance |
| User symbol: | $x_A$ |
| User relation: | $u_{A1}t_{A1} + u_{A1}t_{A2}$ (rule) |
| User components | $[a_{A1}, t_{A1}, u_{A1}, t_{A2}]$ (links) |
| User value: | 219.6 |
| Estimated value: | 219.6 |
| Measurement unit: | km |
| Error Causing Belief: | 1 |
| Local conclusion: | Wrong relation |

*Fig. 3.* Frame structure of a node of the diagnostic network

- The *agenda of targets* is the component through which the engine controls the diagnostic process. The nodes suspected to be fault nodes are inserted in the agenda as pending tasks. The diagnostic mechanism based on appropriate criteria, existing evidences and previous experiences, selects and activates from the agenda a task at a time. Each entry in the agenda includes the following elements: the step of the solution to be examined, the evidential node supporting this entry, the reason of the entry (new step, diagnostic conflict, indirect diagnosis, etc.).
- The *diagnostic knowledge base* of meta-rules. The meta-rules are domain-independent production rules, mainly heuristics, that guide the diagnostic operation. For example, meta-rule N1 (IF there is a new step in the agenda THEN it must be examined at first priority) gives an order of priority for looking through the agenda, while meta-rule N2 (IF there is evidence of malfunctioning THEN the fault will be detected either in the specific evidential node or in its predecessors) assists in the definition of the problem space.
- The *deep-knowledge base.* This contains the correct form of the domain knowledge related to the exercises. The diagnostic mechanism utilises the deep-knowledge base in specific cases such as the diagnosis of students' steps that do not exist in the correct solution (new steps). For example, if the student's step $M_A = D_A V_A$ (where $M_A$ is the mass, $D_A$ the density and $V_A$ the volume of an object $A$), does not exist in the correct solution, then the diagnostic mechanism through the rule:
  IF density $:= D$ and volume $:= V$
  THEN mass $:= M = DV$
  of the deep-knowledge base will acknowledge the correctness of the relation.
- A *partial solver.* This is used for:
  - inferring what output values should be expected for certain input values and,
  - transforming the equation(s) of the expert's solution or the equation(s) of the deep-knowledge base to a form comparable with the equation(s) of the learner's solution.

The diagnostic framework of DIRCE relates the learner's solution to a specific expert's solution identifying the existing discrepancies. All or most of the expert's solutions for a given problem must be known and one of them is selected according to initial evidences acquired from the learner. The basic idea is to use the chosen correct solution as a template during the diagnostic process. The engine based on appropriate evidences, gathered interactively from the learner and the deep-knowledge base, transforms gradually the expert's solution and constructs the learner's solution, diagnosing the existing errors. The operation of DIRCE is strongly based on the *relationship-causality* and the *interactive evidence gathering* approaches:

**Relationship-causality.** The existing relationships among the entities of a procedural network play a significant role in the knowledge representation and problem solving process of a system. The specific diagnostic engine has been built upon the relationship approach in order to avoid the existing difficulties in collecting conditional probabilities and other extra information as well as to bypass complicated and non-real time computationally expensive calculations. The utilisation of the relationship-causality in DIRCE concerns mainly the following three domains:

- *The diagnostic knowledge base*, where many of the production rules refer to the existing relations among the nodes of a diagnostic network.
- *The selection of the most promising node*, where the diagnostic mechanism decides for the appropriate node to be examined in depth.
- *The intermediate diagnosis of the network*, where the local diagnostic results of a node are propagated through its relatives aiming at minimising the diagnostic problem space, at locating areas with other fault nodes, at resolving conflicts, at detecting hidden fault nodes, etc.

Initially, search methods and heuristic algorithms were employed in the relationship identification tasks. These were dependent on the nature and the demands of the diagnostic problems as well as on the structure and the magnitude of the knowledge used. But the phenomena of time consuming operations and demanding implementations, especially in cases of iterated tasks, were noted. In order to surpass these difficulties the Sygg table technique (described in the next section) was developed and used.

**Interactive evidence gathering.** A successful diagnostic procedure is based on the observed symptoms of the malfunctioning system. In many cases only a few manifestations are adequate for drawing proper conclusions (Van der Gaag & Wessels, 1993). DIRCE interactively acquires evidences necessary for the diagnostic operation in two phases. Initially it requests outputs of a preselected set of nodes of the learner's solution. If this information is not adequate, the diagnostic mechanism interactively obtains additional evidences, until a satisfactory diagnosis has been reached.

Diagnostic operation of the method is exemplified in the following chapters by using an exercise from the digital electronics described in Fig. 4. Fig. 5 gives the electronic diagram, while Fig. 6 shows the corresponding diagnostic network of the student's solution.

**PROBLEM** INTEGRATED CIRCUIT ADDERS/MULTIPLIERS

A digital circuit is given consisting of a five adders $A21$, $A23$, $A24$, $A41$, $A42$ and six multipliers $M11$, $M12$, $M13$, $M31$, $M32$, $M33$. The circuit has five inputs with the following values $X1 = 2$, $X2 = 3$, $X3 = 1$, $X4 = 2$ and $X5 = 3$. The question is "What are the outputs of the adders $A41$ and $A42$".

Input $X1 = 2$
Input $X2 = 3$
Input $X3 = 1$
Input $X4 = 2$
Input $X5 = 3$
Multiplier $M11 = X1 \times X2 = 6$
Multiplier $M12 = X2 \times X3 = 3$
Multiplier $M13 = X4 \times X5 = 6$
Adder $A21 = M11 + M12 = 9$
* Adder $A22 = X3 + M13 = 7$
Adder $A23 = X5 + M13 = 9$
Multiplier $M31 = M11 \times A21 = 54$
Multiplier $M32 = A21 \times A22 = 63$
Multiplier $M33 = A22 \times A23 = 26$ (error)
* Adder $A41 = M31 + M32 = 117$
* Adder $A42 = A22 + M33 = 33$ (wrong result)

*Fig. 4.* The digital circuit exercise and a student's solution

## 4. Sygg table

A great number of Artificial Intelligence applications require from their inference mechanisms to detect the existence of direct or indirect relationships between the elements of their knowledge bases. For example, an expert system that diagnoses an integrated circuit with hundreds of gates and a number of outputs needs to know the relationship between a suspected malfunctioning gate and a faulty output. The methods employed, the representation schemes and the heuristic algorithms frequently require a time consuming and demanding implementation in order to accomplish the above request, especially when this occurs many times during an inferential process.

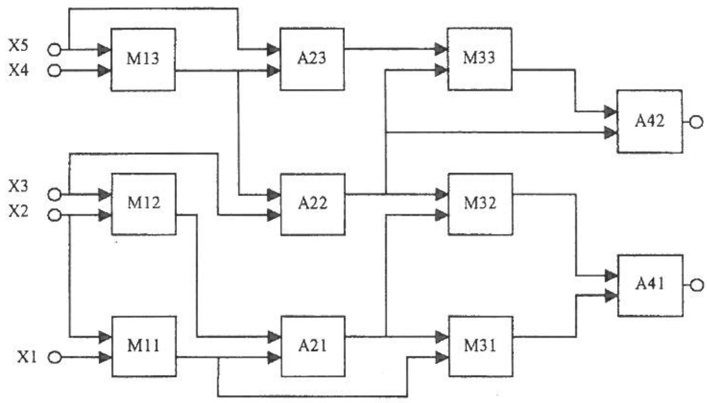


*Fig. 5.* The electronic diagram of the exercise

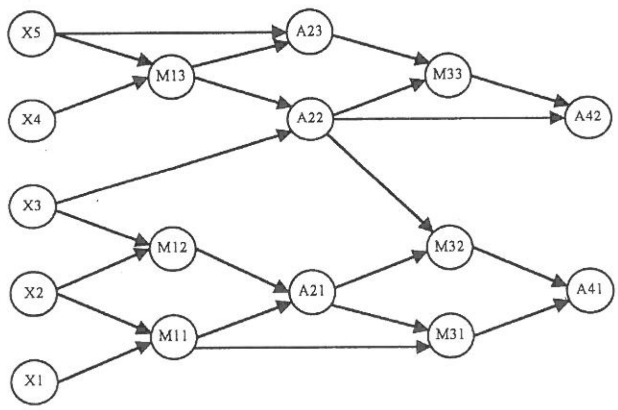


*Fig. 6.* The diagnostic network of the student's solution

The aim of the Sygg table technique, introduced below, is to minimise the processing effort required by the inference mechanism in order to find the existence of any connection among the entities of a knowledge base and/or to minimise the investigated problem space concerning the existing evidences. The term "Sygg" is an abbreviation of the Greek word "syggeneia" which means "relationship".

A Sygg table is a two-dimensional table with identical axes. Each axis of the table is composed of all the entities of the knowledge base. The table is independent from the logic structure of the knowledge base. When new elements are added to the corresponding knowledge base, the table is updated properly. Any number different from '0' in a slot of the table identifies a relationship (direct or indirect) between two entities of the knowledge base. The columns of the Sygg table include the predecessors of the corresponding entities, while the rows contain the descendants of the entities. The crossing point of a row and a column of a specific entity identifies the mark of the entity in the table. The use of this technique in a knowledge representation network can be summarised as follows:

- Relationship identification, detecting instantly any relationship among the nodes of a network.
- Locating the predecessors of a node. Locating the descendants of a node.
- Locating the common intermediate relatives of two or more nodes.
- Detecting the connecting path of two nodes.
- Deducting the diagnostic problem space based on the existing evidences.

The Sygg table in DIRCE is related with the diagnostic network of the solution under examination. The immediate identification by DIRCE of the relationships among the nodes of the network springs from this technique. Each axis of the table is composed of all the nodes of the diagnostic network. Fig. 7 shows the Sygg table of the digital circuit of the previous chapter. If, for example, the engine needs to know which of the two outputs of the digital circuit (adder $A41$ or adder $A42$), is affected by a wrong calculation at multiplier $M12$, it will check the cells $[M12, A41]$ and $[M12, A42]$ of the table (Fig. 7). Their values are 1 and 0, denoting the dependence of $A41$ on $M12$, while the $A42$ is independent.

## 5. Outlining the Diagnostic Reasoning Procedure of DIRCE

The engine initiates the diagnostic process by requesting the outputs of preselected steps of the solution under examination. These steps are preselected by the experts when they prepare the correct solution for the correct solutions data base. DIRCE uses the specific steps as initial evidences to detect whether the solution is erroneous or not. The task is similar to a teacher's action when he/she examines the solution of an exercise that he/she frequently used. At first the teacher examines the steps in which, according to his/her experience, frequently errors occur and then continues with the rest of the diagnostic process.

When all outcomes are correct, the diagnostic procedure terminates. Otherwise, the nodes with the detected discrepancies are recorded as targets for diagnosis in the *agenda of targets*. In the second case, the diagnostic mechanism, based on the learners' answers, selects and uses as template the most plausible correct solution. Diagnosis of the solution under examination is broken down into one or more diagnostic cycles. Each diagnostic cycle begins by selecting a target node from the agenda and it terminates when:

- a fault node is identified with all its faults. The detected faults explain partially or totally the observed discrepancies in the target node,
- the acquired evidence proves the correctness of the target node and justifies the noted discrepancies,
- new faulty nodes are detected and added to the agenda.

By the end of each cycle local diagnostic conclusions are extracted. The character of the diagnostic mechanism is non-monotonic, permitting the alteration of previous diagnostic conclusions based on the recently entered evidential information. Diagnosis arrives at an end when the agenda is empty and all the target nodes have been examined and diagnosed. Each diagnostic cycle is composed of the following stages:

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $A42$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $A41$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| $M33$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| $M32$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| $M31$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| $A23$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| $A22$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| $A21$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| $M13$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| $M12$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| $M11$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| $X5$ | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| $X4$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| $X3$ | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| $X2$ | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| $X1$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| | $X1$ | $X2$ | $X3$ | $X4$ | $X5$ | $M11$ | $M12$ | $M13$ | $A21$ | $A22$ | $A23$ | $M31$ | $M32$ | $M33$ | $A41$ | $A42$ |

*Fig. 7.* The Sygg table of the student's solution

- Selection of the target node.
- Selection of the most promising node.
- Diagnosis of the node.
- Indirect diagnosis of the relative nodes.

It is stressed that DIRCE can diagnose learners' solutions with multiple, mutually dependent fault nodes even though the "one per cycle" fault node identification assumes conditional independence. This is accomplished by requiring the inputs and the corresponding outputs for each interactively examined node. The diagnostic algorithm of DIRCE is shown in Fig. 8. The engine is capable of avoiding:

- Freezing the diagnosis of the solution under examination, meaning the early termination of the diagnostic operation before the extraction of final diagnostic conclusions. This usually happens in the absence of adequate diagnostic evidence.
- Overheating of the diagnostic operation, meaning halting the diagnostic procedure and denoting the inadequacy of the system to manipulate all the available evidences, especially the conflicting ones.

In our example, the initially selected nodes are the adders $A22$, $A41$ and $A42$. The user results are $A22 = 7$ (correct), $A41 = 117$ (correct), $A42 = 33$ (error). Hence, the node $A42$ is inserted in the agenda. The mechanism extracts from the results of the selected nodes the ensuing diagnostic conclusion, "All the nodes of the diagnostic network are correct except $A23$, $M33$ and $A42$ that contain at least one fault".

## 5.1. Selection of the target node

DIRCE selects and activates a target node from the agenda by applying the following criteria:

- the recently inserted new nodes in the network are examined first,
- the nodes with conflicting indirect evidences are examined second,

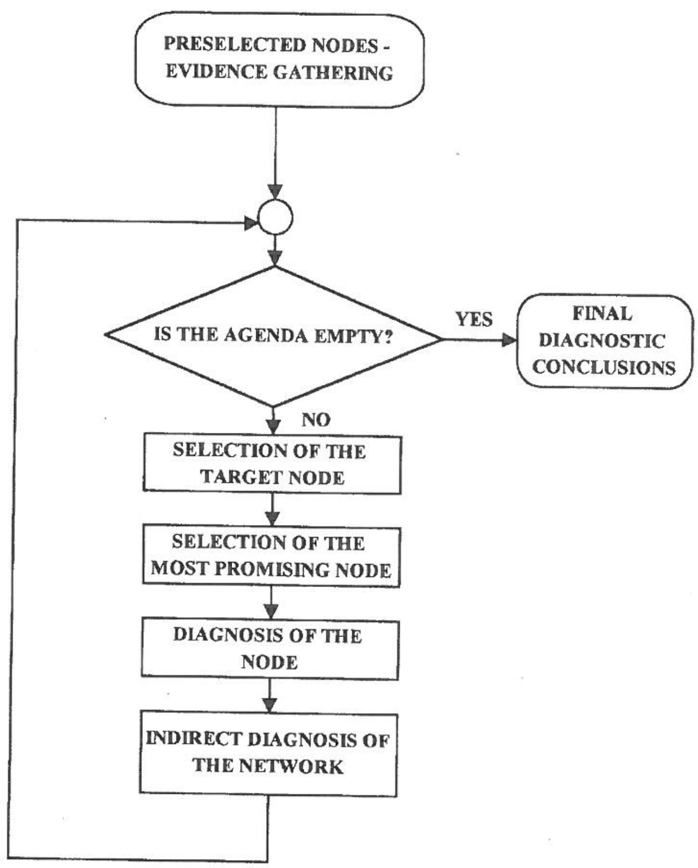


*Fig. 8.* The diagnostic algorithm of DIRCE

• then the nodes with wrong results, and finally,

• the nodes that have not yet been diagnosed in any manner during the diagnostic operation.

The target node is utilised as the main evidence for the detection of the fault nodes as well as for the verification of any examined hypothesis. This node is deleted from the agenda as soon as all the causes of its discrepant behaviour have been diagnosed.

During each cycle the diagnostic mechanism detects and locates the existing errors either in the target node or in one of its predecessors; otherwise, it considers the specific nodes correct. The problem space is determined with the assistance of the Sygg table, by using the target node as well as other existing evidences and the up to that moment extracted conclusions. In the example of the digital circuit, DIRCE selects node $A42$ as target node because of its wrong result and it defines a problem space that involves nodes $A23$, $M33$ and $A42$.

## 5.2. Selection of the most promising node

During this stage the engine is trying to predict and examine the step of the solution that will give the most appropriate evidence for the identification of the error in the defined problem space. The conditional probabilities in the network have only two values (1 or 0), depending on the existence or not of a relationship between the corresponding nodes.

Errors in an individual problem can either be very specific and may be identified in the same step or they may concern a more general misconception and can be identified in different but related steps. Therefore, the fault frequency (prior probability) of each node is not adequate by itself to inform the engine about the most promising node. In order to strengthen the predictability of DIRCE, the Relationship-Causality technique substitutes the prior probabilities with the *error causing beliefs* (ECBs). The error causing belief of a node designates the existing belief for the specific node to be faulty. It can be declared by the expert or evaluated by a machine learning procedure from the previously diagnosed cases or it may be deduced by combining both approaches. The most promising node is chosen according to the following three prerequisites:

• the selected node must be by far the most probable fault node and/or

• the neighbourhood of the selected node must concentrate strong positive evidences that it may include a fault node and/or

• when the particular node is not a fault node, it must radically restrict the relative problem space. In case of lack of further evidence, the probability to detect a fault node in the predecessors or the descendants of the selected node must be roughly equal to 0.5.

Considering the above criteria and the Bayes theorem, a decision support function has been constructed that evaluates the RCF (Relationship Causality Factor). The specific factor is utilised for the selection of the most promising node. The node of the specific problem space with the greater RCF is the requested one. The RCF of a node x is extracted as follows:

$$pp_x = \frac{\sum\limits_{k=1}^{w} p_k + \dfrac{p_x}{2}}{\sum\limits_{i=1}^{n} p_i},$$

where $pp_x$ indicates the relative belief in detecting a fault in node x or among its predecessors, $p_i$, $p_k$ and $p_x$ depict the belief (ECB) of the system that nodes $i$, $k$, $x$ respectively may be the fault nodes,
$w$ is the number of predecessors of $x$ in the specific problem space and
$n$ is the number of all the immediate relatives of node $x$ (including ancestors, descendants and itself) in the area under inspection.

Then $qp_x = |0.5 - pp_x|/0.5$ and $ps_x = \sum\limits_{i=1}^{n} p_i / \sum\limits_{j=1}^{m} p_j$, where $qp_x$ indicates the relative belief deviation according to the third prerequisite, $ps_x$ indicates the relative causal belief of node $x$, and $m$ is the total number of nodes of the problem space, and finally

$$\mathrm{RCF}_x = (1 - qp_x) \times ps_x$$

The values of the $\mathrm{RCF}_x$ range between 0 and 1. When $\mathrm{RCF}_x$ tends to 1, node $x$ is the most promising node. On the other hand, when $\mathrm{RCF}_x$ tends to zero, the specific node does not satisfy one or more of the posed preconditions and it is not recommended. The error causality values and the relationship-causality coefficients of the digital circuit of Fig. 4 are given in Table 1. The most promising node is the multiplier $M33$ with RCF= 0.72.

| | ECB | *Coefficients* | | | |
|---|---|---|---|---|---|
| | | *pp* | *qp* | *ps* | *RCF* |
| $X1$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $X2$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $X3$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $X4$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $X5$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $X5$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $M11$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $M12$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $M13$ | 0.10 | 0.00 | 0.00 | 0.00 | 0.00 |
| $A21$ | 0.30 | 0.00 | 0.00 | 0.00 | 0.00 |
| $A22$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $A22$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $A23$ | 0.24 | 0.24 | 0.52 | 1.00 | 0.48 |
| $M31$ | 0.10 | 0.00 | 0.00 | 0.00 | 0.00 |
| $M32$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $M33$ | 0.16 | 0.64 | 0.28 | 1.00 | 0.72 |
| $A41$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $A42$ | 0.10 | 0.90 | 0.80 | 1.00 | 0.20 |

*Tab. 1.* The RCFs of the digital circuit

## 5.3. Diagnosis of the node

The diagnostic mechanism requires from the user the assigned by the learner relation, the values of the allocated components and the extracted result for the variable under examination. DIRCE, assisted by the partial solver, parses the relation, analyses syntactically and semantically the above elements, diagnoses any existing error in the specific step and arrives at local diagnostic conclusions. If the diagnosed node is a new one in the diagnostic network, the corresponding equation is retrieved from the deep-knowledge base, transformed and finally compared with the learner's equation. When the particular node is not a fault node, there are two directions:

- the acquired information designates the existence of one or more errors among the predecessors of the examined step and the node is inserted in the agenda
- the diagnosed node declares the correctness of its ancestors and delimits the problem space.

In the digital circuit of the example the node $M33$ is, indeed, a fault node with relation $A22 + A23$ instead of $A22 \times A23$ and with the output value equal to 26 instead of the calculated one which is 16 (the correct result is 63).

## 5.4. Indirect diagnosis of the relative nodes

The diagnostic conclusions of the selected and examined node constitute a source of knowledge for the diagnosis of all its relative nodes (predecessors or descendants, with immediate or intermediate relationship). These conclusions are propagated through the neighbouring relative nodes of the diagnostic network by utilising the following two inferential procedures:

**Indirect diagnosis of predecessors.** The diagnostic conclusions are transferred to the predecessors through the parents of the examined node. Any discrepancy between previously extracted results in the nodes and the propelled results indicates a conflict that must be resolved by DIRCE.

**Indirect diagnosis of descendants.** The diagnostic results are diffused sequentially from one level to another, starting from the immediate descendants of the investigated node. The applied process is working in two directions. The initial direction is forward and the knowledge is propagated through the non-examined nodes by recalculating their values. When the forward procedure encounters a previously examined node or a final node, it stops. At each such node the value calculated from the transferred information is compared to the already existing one. When the two values are equal a backward process begins that extracts indirect local diagnostic conclusions for the previously traversed nodes. On the contrary, the appearance of differences declares the existence of one or more fault nodes in the specific problem space, which must be detected by DIRCE.

In the presented example, the indirect diagnosis examines and verifies that node $M33$ is the only cause of the digital circuit malfunction.

## 6. Application of the engine

DIRCE has been applied in the I-DIAS system, which is currently under evaluation, while an initial version of the engine was used in the DIAS diagnostic system (Barbounis & Philokyprou, 1991). I-DIAS can diagnose high school exercises in physics and chemistry with 50 steps at the most. The examined exercises were from kinematics, electrical circuits, heat and inorganic chemistry. All of them had been solved in procedural manner, with steps interconnected through mathematical equations.

The educational objective of I-DIAS is to identify all errors in the students' solutions. The students based on the diagnostic conclusions of the system can proceed to self-correcting actions or to seek the assistance of their teachers if they do not understand why their solutions were wrong.

The diagnostic mechanism of the I-DIAS system can handle adequately all errors that arise in this type of problems with the exception of the covered errors. As covered error is considered any error that cannot be detected via the existing evidences during indirect diagnosis, because a following error covers its discrepant behaviour. The teacher has a similar difficulty in dealing with covered errors when he/she diagnoses a learner's solution based on limited evidence. DIRCE is trying to reveal and identify these errors by examining any new evidence whether it contradicts with any of the up to that moment reached diagnostic conclusions. If there is a conflict, then a covered error has been detected. The engine can diagnose any covered error in a solution when it takes into account all the steps of the student's solution.

The number of diagnostic steps required by the system to locate a fault node in a solution mainly depends on the magnitude of the problem space and on the ECBs of the included nodes. For example, a fault node with very high ECB and/or limited problem space is located in one or two diagnostic steps. On the contrary, the number of diagnostic steps required to locate a fault node with low ECB and/or large problem space approaches $\log_2(N)$, where $N$ is the number of nodes of the specific problem space. The response of the system in consecutive diagnostic steps is nearly instant.

I-DIAS can operate on low and average capabilities microcomputers that usually exist in schools (80386 SX microprocessors and over). It has been developed in Clipper of Computer Associates combined with modules from C libraries. The system can also run in local area networks either under Novell with client-server architecture (Novell 3.1x, Novell 4.x) or under Windows NT of Microsoft in DOS mode. Furthermore, I-DIAS with the assistance of the communication package PC-Anywhere of Symantec can run remotely through common telephone lines. In the client-server operation of the system a work area is created at the client's workstation during the diagnostic process. This area includes the diagnostic network, the metarules knowledge base and the agenda of targets. The program communicates with the server in order to query the deep knowledge base only. By the end of the diagnostic process the working area is eliminated and the final conclusions update proper data bases in the server.

## 7. Conclusions

The present paper was focused on the intelligent diagnostic engine DIRCE. The engine examines problems exclusively solved in a procedural manner, with steps interconnected through mathematical equations. The following characteristics and abilities, suitable to distance diagnosis, have been embodied in DIRCE:

- applicability to a wide range of knowledge domains,
- interactive diagnosis of the student's solutions in an evidence-driven approach, requiring only the necessary information from the user,
- identification of the learner's errors without extensive bug catalogues and mal-rule libraries,
- perception and detection of the existing errors in a manner efficient and rapid,
- capability to operate in multi-user environments.

DIRCE is currently being used and tested in the I-DIAS system for diagnosis of high school exercises in physics and chemistry with positive results. Main objective of the system is the online identification of the existing errors in the students' solutions. During informal tests the engine based on limited information acquired interactively from the students, located all the existing errors of the diagnosed solutions, except for the covered errors.

Future research will be concentrated at minimising the number of required diagnostic steps in students' solutions with average or low belief nodes as well as at minimising the information required from the experts for the data and knowledge bases of the DIRCE systems.

## References

ANDERSON J. R., Production Systems, Learning and Tutoring. In: Production System Models of Learning and Development, The MIT Press, (1988), 437–458.

ANDERSON J. R., BOYLE C. F., CORBETT A. T. AND LEWIS M. W., Cognitive modeling and intelligent tutoring. Artificial Intelligence, 42, (1990), 7–49.

BARBOUNIS E. AND PHILOKYPROU G., Analysis of the students' responses — ITS in diagnosis of the pupils' learning level, (In Greek). In: Proc. 3rd Hellenic Conference in Information Technology, Vol. 2, (1991), 409–421. Athens, Greece.

BARBOUNIS E. AND PHILOKYPROU G., Distance diagnosis of erroneous students' solutions by an intelligent diagnostic system, (In Greek). In: Proc. 2nd PanHellenic Conference in Information Technology and Tutoring Mathematics, (1995), 681–699. Nicosia, Cyprus.

BURTON R. B. AND BROWN R. B., An investigation of computer coaching for informal learning activities. In: D.H.Sleeman & J.S.Brown (Eds.), Intelligent tutoring systems, New York: Academic Press, (1982), 79–98.

CLANCEY W. J., Tutoring rules for guiding a case method dialogue. In: D.H.Sleeman & J.S.Brown (Eds.), Intelligent tutoring systems, New York: Academic Press, (1982), 201–225.

FOSTER J., International telecomputing. In: R.F.Tinker and P.M.Kapiskivsky (Eds.), Prospects for Educational Telecomputing: Selected Readings. TERC, Cambridge, Mass, (1992).

HUNTER B., Internetworking: coordinating technology for systemic reform. Communications of the ACM, Vol.36, No.5, (1993), 42–46.

KEUNG–CHI NG. AND ABRAMSON B., Uncertainty Management in Expert Systems, IEEE Expert, April 1990, (1990), 29–47.

KEYES J., Why expert systems fail. AI Expert, Nov. 1989, 50–53.

LANGLEY P., WOGULIS J. AND OHLSSON S., Rules and principles in cognitive diagnosis. In: N.Frederiksen, R.Glaser, A.Lesgold and M.G.Shafto (Eds.), Diagnostic monitoring of skill and knowledge acquisition, Hillsdale, NJ, Elbraum, (1990), 217–250.

OHLSSON S. AND LANGLEY P., Psychological evaluation of path hypotheses in cognitive diagnosis. In: H.Mandl & A.Lesgold (Eds.), Learning issues of intelligent tutoring systems. Springer–Verlag, New York, (1988), 42–62.

OHLSSON S., Impact of cognitive theory on the practice of courseware authoring. Journal of Computer Assistant Learning, 9, (1993), 194–221.

PAYNE S. J. AND SQUIBB H. R., Algebra mal-rules and cognitive accounts of errors. Cognitive Science, 14, (1990), 445–481.

PEARL J., A constraint-propagation approach to probabilistic reasoning. In: Uncertainty in Artificial Intelligence, Vol. 1, Elsevier Science Publishers B.V. (North Holland), (1986), 357–369.

PEOT M. A. AND SHACHTER R. D., Fusion and propagation with multiple observations in belief networks. Artificial Intelligence, 48, (1991), 299–318.

VAN DER GAAG L. AND WESSELS M., Selective Evidence Gathering for Diagnostic Belief Networks, AISBQ 86, Winter 93/94, (1993), 23–34.

VANLEHN K., Student Modeling. In: M.Polson & J.Richarson (Eds.), Foundation of Intelligent Tutoring Systems, 3(2), Lawrence Elbraum Associates, (1988), 55–77.

*Received:* October, 1995
*Accepted:* October, 1996

*Contact address:*
Manolis Barbounis
Filikis Etairias 6
185 39, Pireaus, Greece
Telephone: +301 80 94 055
Fax: +301 80 94 050
E-Mail: ieim7@netor.gr

EMMANUEL BARBOUNIS is doctoral student in the Department of Informatics at the University of Athens. He received the B.Sc degree in physics from the University of Athens and the M.Sc. degree in computing systems from the Cranfield Institute of Technology. His main research interests are in expert systems, intelligent diagnosis and their application in education.

MARIA GRIGORIADOU is Assistance Professor in the Department of Informatics at the University of Athens. She received the B.Sc degree in physics from the University of Athens, the D.E.A degree in "automatique theorique et theorie de traitement de l' information" and the Ph.D degree in computer science both from the University of Paris VII. Her current research interests are in the development of multimedia environments, Intelligent Tutoring Systems and Interactive Learning Environments.

GEORGE PHILOKYPROU is Professor in the Department of Informatics at the University of Athens. He received the B.Sc degree in physics and the M.Sc degree in electronics from the University of Athens, and the Ph.D degree in Computer Science from the University of Manchester. His current research interests include database systems, computer architecture, parallel processing, natural language processing and intelligent educational systems.