

AN AUTHORIZING TOOL FOR ALGEBRA-RELATED DOMAINS

Maria Virvou, Maria Moundridou

Department of Computer Science, University of Piraeus,
80, Karaoli and Dimitriou St., Piraeus 185 34, Greece
mvirvou@unipi.gr, mariam@unipi.gr

1 Overview

This paper describes the work-in-progress of the development of an authoring tool for Intelligent Tutoring Systems (ITS). The main objective of this tool is to be useful to teachers and students of domains that make use of algebraic equations. Such domains could be chemistry, economics, medicine, physics etc. The initial input to the tool will be a “description” of a specific domain given by a human teacher. In return the tool will provide assistance at the construction of exercises by the human teacher. Then the authoring tool will also be able to monitor the students while they are solving the exercises and provide appropriate feedback.

Although ITSs can provide quite powerful interactive learning environments most of them are research products that have not been used extensively in real classrooms. ITSs have often been criticised that they miss the mark in terms of task reality, feasibility and effectiveness (Mc Graw, 1994). One reason for this has been the difficulty in developing an Intelligent Tutoring System even in small domains. For example, Woolf and Cunningham (1987) have estimated that the development of an ITS takes more than 200 hours to produce an hour of instructional material, which in most cases cannot be re-used. A solution to these problems may be the development of authoring tools which will help construct cost-effective and reusable ITSs in various domains.

It has been generally acknowledged (Hartley & Sleeman, 1973; Burton & Brown, 1976; Wenger, 1987) that the main components of an ITS are the domain knowledge, the student modelling component, the advice generator and the user interface. Accordingly, there are authoring tools that focus on various aspects concerning each of these components. An example of an authoring tool

dealing with the representation of instructional expertise is REDEEM (Major et al., 1997). REDEEM expects the human instructor to categorise tutorial “pages” in terms of their difficulty, their generality, and whether they are prerequisite for other pages. Other systems such as RIDES (Munro et al., 1997) provide tools that can generate instruction. RIDES is used for the construction of tutoring systems that teach students how to operate devices through simulations. A system which adds capabilities to RIDES is DIAG (Towne, 1997). DIAG is concerned with the creation of domain knowledge by providing a mechanism that is applicable to many domains which are related to fault diagnosis. Similarly, the authoring tool described in this paper, incorporates knowledge about the construction of exercises and a mechanism for student error diagnosis that is applicable to many domains that make use of algebraic equations.

2 Description of the system

In particular the tool takes input from a human instructor about a specific equation-related domain (e.g. economics). This input consists of knowledge about variables, units of measure, formulae and their relation. The instructor does not have to provide the complete list of variables and equations which describe the domain, all at once. S/he may only enter the ones that will be used in the problems to be constructed in the current interaction and add more in subsequent interactions. The tool accumulates domain knowledge each time that the human instructor gives new input. This means that the instructor may give information to the tool at the same rate as lessons progress in a course. An example of input to the system that an instructor could provide from the domain of physics is shown in Table 1.

Table 1: Example of input to the system from the domain of physics

Variables:	
Velocity	v
Initial velocity	v_0
Acceleration	a
Time	t
Force	F
Mass	m
Impulse	J
Formulae:	
F=m*a	
J=m*v	
$v=v_0+a*t$	

A second example from the domain of economics which is also equation related can be seen in Table 2.

Table 2: Example of input to the system from the domain of economics

Variables:	
Income	Y
Consumption	C
Investment	I
autonomous consumption	a
marginal propensity to consume	b
autonomous investment expenditure	I
Interest rate	R
sensitivity of investment in interest rates	c
Formulae:	
$Y=C+I$	
$C=a+b*Y$	
$I=i-c*R$	

2.1 Instructor's mode

When the human instructor wishes to create exercises s/he can type in what is given and what is asked and the tool can either construct the full problem text or provide consistency checks that help the instructor verify its completeness and correctness. In case of redundancies in the given data the tool lets the instructor know. After the construction of a problem the tool lets the instructor preview the problem text and the solution of the exercise as formulated by the system. The tool incorporates a user modelling mechanism that focuses on diagnostic reasoning about students' errors. This mechanism will be used by the system when the problem that the instructor has created is tackled by a student in the student mode.

There are two types of problem that the system can assist the instructor to construct:

Problems without numbers. In problems without numbers the system displays every variable that the human instructor has entered. The human instructor should specify which variable is the unknown, which one is given and the type of change. For example in the domain of economics the instructor could select as unknown the variable "income", and as given an "increase" at the level of

“interest rates”. The system would then produce the following problem text: “How will the increase of interest rates affect the level of income?”.

Problems with numbers. In problems with numbers the system displays again every variable that the human instructor has entered and requests the unknown. The system considers automatically all the variables which depend on the “unknown” (according to the equations) as possible given data. These variables are shown to the instructor who should now enter their values. An example is illustrated in Figure 1. The system follows the instructor’s actions and reports any inconsistencies. For example, if the instructor enters values for both the dependent and independent variables of an equation the system points out the error. Finally, the system produces the problem text. An example of problem text is the following: “If the force is 100 Newtons, the mass is 25 kgr, the initial velocity is 0 m/sec² and the time is 5 secs, then find the impulse.”. The instructor may change the problem text to make it more comprehensible; for example: “A force of 100 Newtons is acting on a 25 kgr object which is initially stable. After 5 secs how much is the impulse?”.

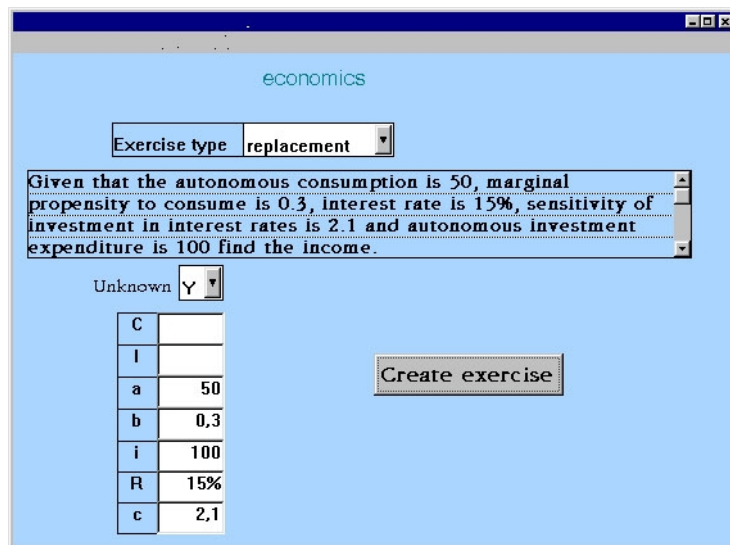


Figure 1: A sample screen

2.2 Student’s mode

When a student attempts to solve an exercise the system provides an environment where the student gives the solution step by step. The system

compares the student's solution to its own. The system's solution is generated by the domain knowledge about algebraic equations. While the student is in the process of solving the exercise the system monitors his/her actions. If the student makes a mistake, the diagnostic component of the system will attempt to diagnose the cause of it. The diagnosis of the underlying cause of a mistake is a difficult task for an ITS because the observable symptoms need to be analysed further. As Hollnagel (1991; 1993) points out, there is an important distinction between the underlying cause or genotype of an error and the observable manifestation or phenotype of the error.

There are three categories of causes that can be recognised by the system:

Typographic errors. In this case, errors are ignored as far as the student knowledge is concerned.

Mathematical errors. In this case, errors are due to solving equations incorrectly.

Domain errors. In this case, errors are due to lack of knowledge or misconceptions concerning the domain taught. For example, the student may not remember correctly one formula.

Each student error is attributed to one of the three categories listed above. For example, if a student types in the equation: $x+5= = 8$, where there has been a repetition of the equal sign, the system recognises this as a typographic error and ignores it. However, there may be cases where a certain error may be attributed to more than one category. For example, if a student types in the equation $x+5=9$ where 9 should have been 8, then this error could either be a typographic error or a mathematical error. In cases like this the system uses a history mechanism about a particular user in order to resolve ambiguities. For example, if the particular user has been recorded to have frequently made typographic errors but no mathematical errors at all then the system may favour the hypothesis of a typographic error in this case too.

The "history mechanism" records certain student features that have been inferred during past interactions such as persistence of a certain type of error (e.g. mathematical error). These features form the long term student model (Rich, 1979; 1983) which is used by the system in two ways:

- to resolve ambiguities that arise from errors for which more than one hypothesis can be generated as to what the cause of the error has been.
- to form individualised progress reports of the student which could be requested by the student and/or the instructor.

3 References

Burton, R.R. & Brown, J.S. (1976). A tutoring and student modelling paradigm for gaming environments. *Computer Science and Education*, Colman, R. and Lorton, P.Jr. (eds.), ACM SIGCSE Bulletin, Vol. 8, no. 1 pp. 236-246.

Hartley, J.R., Sleeman D.H. (1973). Towards intelligent teaching systems. *Int. Jl. Man-Machine Studies*, Vol. 5, pp. 215-236.

Hollnagel, E. (1991). The Phenotype of Erroneous Actions: Implications for HCI Design. *Human-Computer Interaction and Complex Systems*, G. R. S. Weir and J.L. Alty (eds.), London Academic Press Ltd.

Hollnagel, E. (1993). The Phenotype of Erroneous Actions. *International Journal of Man-Machine Studies*, Vol. 39, pp. 1-32.

Major, N., Ainsworth, S. & Wood, D. (1997). REDEEM: Exploiting Symbiosis Between Psychology and Authoring Environments. *International Journal of Artificial Intelligence in Education*, 8, 317-340.

Munro, A., Johnson, M., Pizzini, Q., Surmon, D., Towne, D. & Wogulis, J. (1997). Authoring Simulation-centered tutors with RIDES. *International Journal of Artificial Intelligence in Education*, 8, 284-316.

Rich, E. (1979). User Modelling via Stereotypes. *Cognitive Science*, 3 (4), 329-354.

Rich, E. (1983). Users as Individuals: Individualizing User Models. *International Journal of Man-Machine Studies*, Vol. 18, 199-214.

Towne, D. (1997). Approximate reasoning techniques for intelligent diagnostic instruction. *International Journal of Artificial Intelligence in Education*, 8, 262-283.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. Morgan Kaufman.

Woolf, B.P. & Cunningham, P.A. (1987). Multiple knowledge sources in intelligent teaching systems. *IEEE Expert*, Summer 1987.