http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 7 (3): 528-532, 2008
ISSN 1812-5638

JEE-Tutor: An Intelligent Tutoring System For Java Expressions Evaluation

Samy S. Abu Naser
Faculty of Engineering and Information Technology, Al-Azhar University, Gaza, Palestine

Abstract: This study presents an Intelligent Tutoring System called JEE-Tutor for teaching Java operator precedence, associativity and expressions evaluation to freshman students in the Faculty of Engineering and Information Technology in Al-Azhar University. An overview of the JEE-Tutor architectural design and user interface will be discussed. According to the success of other similar Intelligent Tutoring Systems, it is also hypothesized that students will be able to learn operator precedence, associativity, expressions evaluation and gain knowledge more quickly and effectively than students using traditional methods of teaching.

Key words: Artificial intelligence, tutor, intelligent tutoring system, E-learning, java programming

INTRODUCTION

An Intelligent Tutoring System (ITS) provides an individualized computer-based instruction to students (Abu Naser, 2006, 2001; Brusilovsky *et al*., 1996). These systems emerged from application of artificial intelligence techniques to the Computer Aided Instruction (CAI) systems (Abu Naser and Sulisel, 2000). The difference is that an ITS usually compares the student's work with expert solutions or strategies, models the student's probably knowledge of a domain and provides coaching or advice, taking into account what the student's knowledge state and preferred learning style. Depending on Artificial Intelligence (AI) and cognitive science, ITS became very popular and effective domain in Education for many reasons: Better student performance, student learn in less time and student is in the driver seat (Anderson *et al*., 1995; Woolf *et al*., 2001; Graesser and Person, 2001). For many years, there is a continuous development and evaluation of ITS (Shute, 1995; Shute and Glaser, 1990) for tutoring and monitoring programming in the field of Artificial Intelligence in Education. Programming requires a group of problem-solving and diagnostic strategies. The behavior in which a student writes code provides great insight into the way of his thinking. As a result, programming provides attractive area to study learning and cognitive processes (Koedinger, 2001; Scott *et al*., 1991). Among the objectives of this research is to gather the developments in the ITS, Cognitive Science and AI to make a useful intelligent tutor to help students understand Java programming expressions evaluation.

Usually textbooks do not present the steps required to solve a problem, but using visualizations (Abu Naser, 2008) and intelligent tutoring systems, student can learn and solve problems comfortably.

Architecture and design of JEE-tutor: We present the model and architecture for the Java expressions evaluation intelligent tutoring system, the knowledge base design, expert module design, feedback module design and the user interface design.

The JEE-Tutor could be part of a courseware on teaching Java programming language to undergraduate freshman students at the university-level.

JEE-Tutor knowledge base design: Here, we describe the Knowledge Base architectural model for JEE-Tutor. In this study, we have concentrated on the syntax of expressions, operator precedence, associatively and evaluation of Java expressions.

Java has well-defined rules for specifying the order in which the operators in an expression are evaluated when the expression has several operators (Liang, 2008; Thomas, 2006; Horstmann, 2006).

Precedence order: When two operators share an operand the operator with the higher precedence goes first. For example, 1+2 * 3 is treated as 1+(2*3), whereas 1*2+3 is treated as (1*2)+3 since multiplication has a higher precedence than addition.

Associativity: When two operators with the same precendence, the expression is evaluated according to its associativity. For example x = y = z = 13 is treated as x = (y = (z = 13)), leaving all three variables with the value 13, since the = operator has right-to-left associativity (and an assignment statement evaluates to the value on the right hand side). On the other hand, 60/2/3 is treated as (60/2)/3 since the/operator has left-to-right associativity.

We have stored the operators of Java, its precedence and associativity in the knowledge base (Fig. 1).

Presedence	Operator	Type	Associtivity
13	( )	Parentheses	Left to right
12	+ + - -	Unary post-increment Unary post-decrement	Right to left
11	+ + - - + - ! ~	Unary pre-increment Unary pre-decrement Unary plus Unary minus Unary logical negation Unary bitwise complement	Right to left
10	* / %	Multiplication Division Modulus	Left to right
9	+ -	Addition Subtract	Left to right
8	< <= > >=	Relational less than Relational less than or equal Relational greater than Relational greater than or equal	Left to right
7	= = ! =	Relational is equal to Relational is not equal to	Left to right
6	&	Bitwise AND	Left to right
5	^	Bitwise exclusive OR	Left to right
4	|	Bitwise inclusive OR	Left to right
3	& &	Logical AND	Left to right
2	||	Logical OR	Left to right
1	= + = - =	Assignment Addition assignment Subtarction assignment	Right to left

Fig. 1: Operator precedence and its associativity

For every problem in the knowledge base, we have stored the possible solution, possible errors for specific categories and possible hints for each error. Figure 2 shows a problem example with a solution and some errors.

The problem stored in the knowledge base can be classified into 5 levels of difficulties. Difficulties mean more operators, mixed associativity types and different level of precedence, are involved in the problem. The Expert model determine which problem level should be given to the student.

In order for the student to solve a problem, he must first determine operator order of evaluation by filling the table as shown above. Second, the student should fill the second table by evaluating each operator at a time and third, put the output of the expression in the last line of the second table as shown in Fig. 2.

JEE-Tutor expert module: JEE-Tutor expert module is an expert system that is used to determine the student level of understanding from the problem statement, the problem specification and student's answers. When the student reaches a certain score answering at the current level of difficulty; for example 80% or more, the expert system

Problem: Evaluate the expression ex 1 in the following Java program segment and find the out put of the program

```
class Precedence Ex1 {
  public static void main (String [ ] args) {
  int a = 6;
  int  b = 5;
  int c = 10;
  int d = 8;
  float ex 1 = 0;
  ex1 = a% b* ++c + –d* 2);
 }
}
```

The solution which is authored by Instructor:

Operator	=	%	*	++	-	--	*
Evaluation order	7	3	4	2	6	1	5

Operation	Operation result	Expression
1	7	–d
2	11	++c
3	1	a % b
4	11	a % c * ++c
5	14	–d * 2
6	25.0	a % c * ++c + –d * 2
7	25.0	ex1 = a % c * ++c + –d * 2
Output of the Program		**The result is: 25.0**

The student first possible error: Associativity problem between ++b and --c

Operator	=	%	*	++	-	--	*
Evaluation order	7	4	3	1	5	2	6

The student second possible error: Associativity problem between % and *

Operator	=	%	*	++	-	--	*
Evaluation order	7	4	3	1	5	2	6

The student third possible error: Precedence problem between - and +

Operator	=	%	*	++	-	--	*
Evaluation order	7	4	3	1	5	2	6

Fig. 2: A problem example with a solution and a few possible errors

increases the level of difficulties of problems to be given for the student. On the other hand, if the student did not reach a minimum score at certain level say 50%, the expert system branches the student to the tutorial on the subject involved in the problem. When the student finishes the tutorial, the expert system permits the student to go back to the questions mode. The score of the student, the level of difficulties and his name are all stored in a database for further analysis in the future when the student comes back and use the JEE-Tutor. So, the database reflects the actual level of every student as a result of all previous sessions.

JEE-Tutor feedback module: Effectiveness of the system depends heavily upon its feedback timing and style. Timing refers to when the student is given a response to the solution. When the feedback is presented to the

Student solution

Operator	=	%	*	++	-	–	*
Evaluation order	7	4	3	1	5	2	6

JEE-Tutor: "There is an associativity problem with ++ and –.
The order of evaluation is Right to left". Do you remember that rule?"
Student: Yes

JEE-Tutor makes the correction and proceeds with rest of the operators table

Operator	=	%	*	++	-	–	*
Evaluation order	7	4	3	2	5	1	6

JEE-Tutor: "There is an associativity problem with % and *-.
The order of evaluation is Left to rifgt". Do you remember that rule?"
Student: Yes

JEE-Tutor makes the correction and proceeds with rest of the operators table

Operator	=	%	*	++	-	–	*
Evaluation order	7	3	4	1	5	2	6

JEE-Tutor: "There is an associativity problem with - and *-.
The precedence of * is higher than -". Do you remember that rule?"
Student: Yes

JEE-Tutor makes the correction and student continue with the solution of the problem by filling the second table

Fig. 3: A dialogue between JEE-Tutor and the student

student should be governed by what the student have done. Tutors are better than teachers in this respect (Du Boulay, 2000; Mark and Greer, 1995) in that they can provide a student with timely feedback better than most teachers.

Using the example in Fig. 2, the following dialogue between JEE-Tutor and the student would come up as in Fig. 3.

JEE-Tutor user interface: The interface of intelligent tutoring systems is a very important factor that we gave it a careful consideration during the design of JEE-Tutor. The user interface is based on a presentation format implemented in many popular Integrated Development Environments used by professional programmers (Conlan *et al*., 2002). Upon connecting to JEE-Tutor website, the student's browser displays the working environment for JEE-Tutor An appropriate skill-level problem is selected or the problem that last attempted is presented to the student.

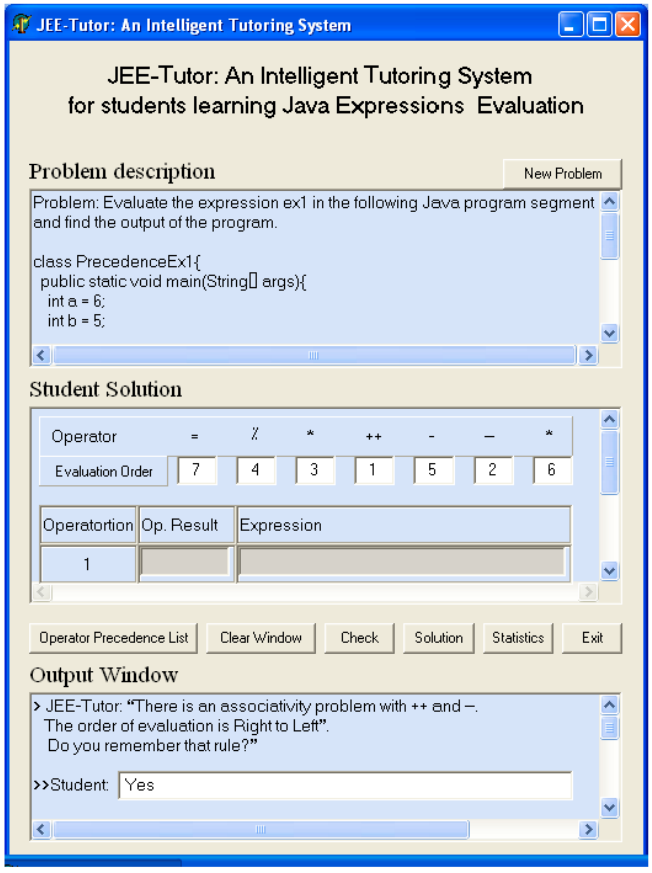

Fig. 4: JEE-Tutor user interface

The student solves the problem in the Student Solution window. Once the student fills the operators evaluation order table and he press the Check button, the expert module will determine the appropriate response based on the diagnosis of the student solution. A dialogue will be initiated between the JEE-Tutor and the student in case of errors. The expert system module informs the user about each error and ask the user weather he remember the rule of precedence or associativity for that operator. If the student answers indicates that he does not remember that rule, the expert system using pattern matching template analyze the student response and display the operator precedence table. Before the expert module proceeds with the next error, it corrects the previous error for the student. Once all errors are corrected, the user will be given control to fill the table of expression evaluation to find its final answer.

The intelligent feedback of the expert module is sent to the student's output window. The student, at any time, may explicitly request from JEE-Tutor to view the solution, Exit from the current problem and ask for a new one; furthermore, the student can view his performance based on statistics including problems attempted, problems solved, number of attempts on a problem and problem difficulty. The JEE-Tutor user interface is shown in Fig. 4.

CONCLUSIONS

In this study, we have presented recent developments related to the JEE-Tutor Intelligent Tutoring System for teaching students Java Expressions Evaluation. JEE-Tutor is based on sound theories, pattern recognition techniques, error detection and correction strategies as in (Sykes and Franek, 2003; Aleven and Koedinger, 2002). Even though an evaluation of the JEE-Tutor has not been done yet, the implemented examples discussed clearly demonstrates the potential of JEE-Tutor. This research is significant since it has the potential to be applied to many programming courses at the university-level.

FUTURE WORK

JEE-Tutor has not yet been evaluated as to its effectiveness as a tutoring tool. A full evaluation is planned during 2008, to be taken with an introductory Java programming class. Additional intelligent tutoring systems for Java programming should be invested to enable students overcome their difficulties that are faced in first programming languages.

REFERENCES

Abu Naser, S.S. and O. Sulisel, 2000. The Effect of using computer aided instruction on performance of 10th grade biology in Gaza. J. Coll. Educ., 4 (1): 9-37.

Abu Naser, S.S., 2001. A comparative study between animated intelligent tutoring systems AITS and video-based intelligent tutoring systems VITS. Al-Aqsa Univ. J., 5 (1): 72-96.

Abu Naser, S.S., 2006. Intelligent Tutoring System (ITS) for teaching database to sophomore students in Gaza and its effect on their performance. ITJ., 5 (5): 916-922.

Abu Naser, S.S., 2008. Developing visualization tool for teaching AI searching algorithms. ITJ., 7 (2): 350-355.

Aleven, V. and K.R. Koedinger, 2002. An effective metacognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor. Cognitive Sci., 26 (2): 147-179.

Anderson, J.R., A.T. Corbett, K.R. Koedinger and R. Pelletier, 1995. Cognitive Tutors: Lessons learned. JLS., 4: 167-207.

Brusilovsky, P., E. Schwarz and G. Weber, 1996. ELM-ART: An Intelligent Tutoring System on World Wide Web. In: Intelligent Tutoring Systems, Claude Frasson, Gilles Gauthier and Alan Lesgold (Eds.). Volume 1086 of Lecture Notes in Computer Science, Berlin, Germany. Springer Verlag, pp: 261-269,

Conlan, O., C. Hockemeyer, V. Wade, D. Albert and M. Gargan, 2002. An architecture for integrating adaptive hypermedia service with open learning environments. In: Proceedings of ED-MEDIA 2002, Volume 1 of World Conference on Educational Multimedia. June 24-29, 2002. Hypermedia and Telecommunications, pp: 344-350.

Du Boulay, B., 2000. Can we learn from ITSs? In: Intelligent Tutoring Systems. Proceedings of the 5th International Conference ITS, Gauthier, G., C. Frasson and K. VanLehn (Eds.). Springer-Verlag, Berlin, 1839: 9-17.

Graesser, A.C. and N.K. Person, 2001. Teaching tactics and dialog in auto tutor. IJAIE., 12: 12-23.

Horstmann, 2006. Big Java. 2nd Edn. John Wiley and Sons.

Koedinger, K.R., 2001. Cognitive Tutors. In: Smart Machines in Education, Forbus, K.D. and P.J. Feltovich (Eds.). Cambridge, MA: MIT Press, pp: 145- 167.

Liang, D.Y., 2008. Introduction to Java Programming-Comprehensive Version. 7th Edn. Prentice Hall.

Mark, M. and J.E. Greer, 1995. The VCR tutor: Effective instruction for device operation. J. Learn. Sci., 4 (2): 209-246.

Scott, A.C., J.E. Clayton and E.L. Gibson, 1991. A Practical Guide to Knowledge Acquisition. (Menlo Park, CA: Addison-Wesley).

Shute, V. J. and R. Glaser, 1990. A large-scale evaluation of an intelligent discovery world: Smithtown. Interactive Learn. Environ., 1: 51-77.

Shute, V.J., 1995. SMART Evaluation: Cognitive Diagnosis, Mastery Learning and Remediation. In: Proceedings of AI-ED 95, Charlottesville. 16-19 August 1995. Greer, J. (Ed.). VA: AACE, pp: 123-130.

Sykes, E.R. and F. Franek, 2003. A prototype for an intelligent tutoring system for students learning to program in Java. Proceedings of the IASTED International Conference on Computers and Advanced Technology in Education. June 30-July 2, 2003. Rhodes, Greece, pp: 78-83.

Thomas, W.U., 2006. An Introduction to Object-Oriented Programming with JAVA, 4/e Tata McGraw-Hill Publications.

Woolf, B.P., J. Beck, C. Eliot and M. Stern, 2001. Growth and Maturity of Intelligent Tutoring Systems. In: Smart Machines in Education, Forbus, K.D. and P.J. Feltovich (Eds.). Cambridge, MA: MIT Press, pp: 100- 144.