EXTENSIONOF AN INTELLIGENT TUTORING SYSTEM FOR DEBUGGING

Authors:

Joyce Ann D. Rada
Javelin Karl C. Magtalas
Carlo T. Martinez

Affiliation:

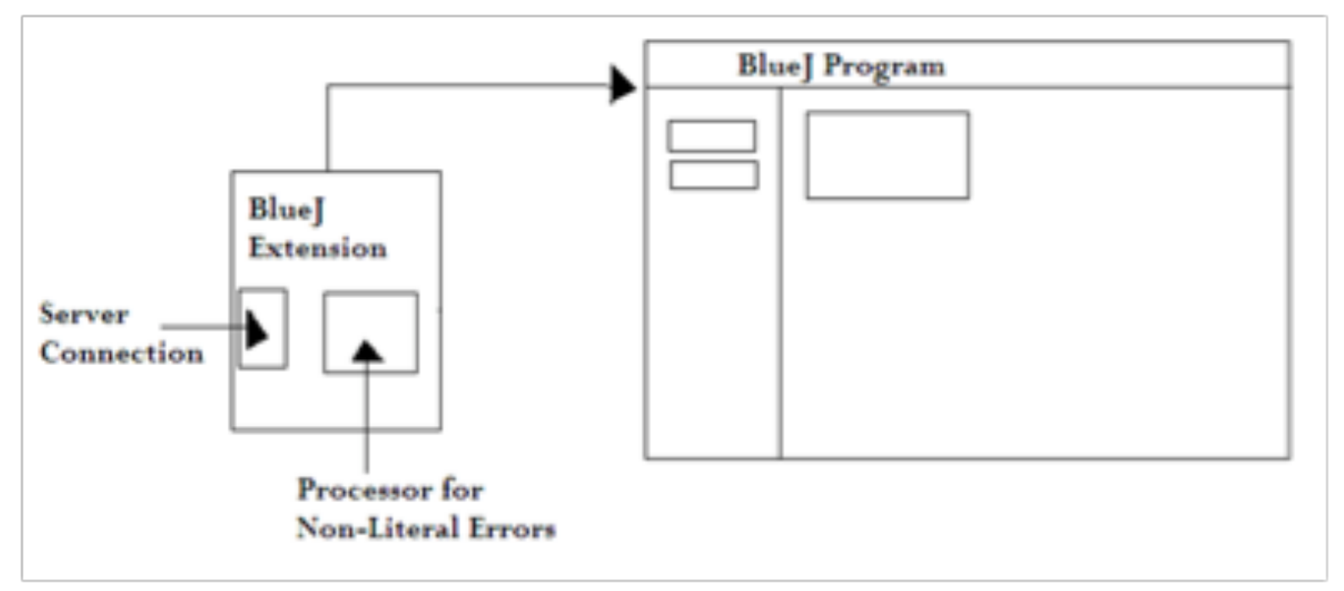
Ateneo Laboratory for the Learning Science,
Department of Information Systems and Computer Science,
Ateneo De Manila University

INTRODUCTION

Debugging is the process of locating errors in source program and fixing them. Errors as reported by the compiler may be classified as literal or non-literal. Literal errors are syntactic errors correctly identified with compiler errors messages. Non-literal errors, on the other hand, are errors that are incorrectly reported by compiler error messages. In this light, the intelligent tutoring system for novice programmers has been developed to aid novice programmers in debugging non-literal errors. The intelligent tutoring system will serve as a guiding tool for novice programmers by showing the proper error message, an explanation of the nature of the error, and an accompanying pair of syntactically correct and incorrect sample code. With novice programmers as the target audience, we improved the interaction between them and the intelligent tutoring system by simplifying the graphical user interface and providing a deeper analysis of the log collected by the server. We also integrated the system to the IDE BlueJ as an extension. We believe that this made debugging non-literal errors for novice programmers an easier experience.

METHODOLOGY

SYSTEM REDESIGN



The BlueJ extension created by the previous group was used as a reference by our group for the further evaluation and improvement of the system. It sill looked for errors and continued recording logs everytime there's a user input.

ITS SOURCE CODE REVIEW

The source code of the ITS is composed of sixteen (16) classes and one (1) interface

Interface	Messages
ErrorHandler	Tools
Classes	CodeScanner
BraceExpectedErrorHandler	SampleForm
IncompatibleTypesErrorHandler	Processor
SemicolonExpectedErrorHandler	Script
CannotFindSymbolErrorHandler	ScriptReader
BracketExpectedErrorHandler	SimpleExtension
MissingReturnStatementErrorHandler	ITSForm
IllegalStartOfStatementErrorHandler	

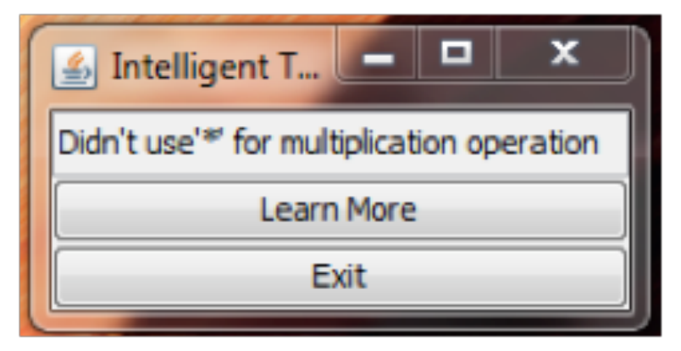
The interface, 'ErrorHandler', is an interface that provides two (2) methods to be defined by implementing classes. These two methods are 'boolean accepts (String errorMessage)' and 'String process (int lineNumber, String code)'. The first method is designed to check if the errorMessage parameter passed in belongs to the ErrorHandler class implementing this method. It will return a value of true if the errorMessage does belong to this certain class and false if otherwise. The second method processes the error of this class type given a lineNumber and the code. It then returns the appropriate String message depending on the actual error of this category. Seven classes implement this interface

Table 2. Class Implementing "Errorhandler"

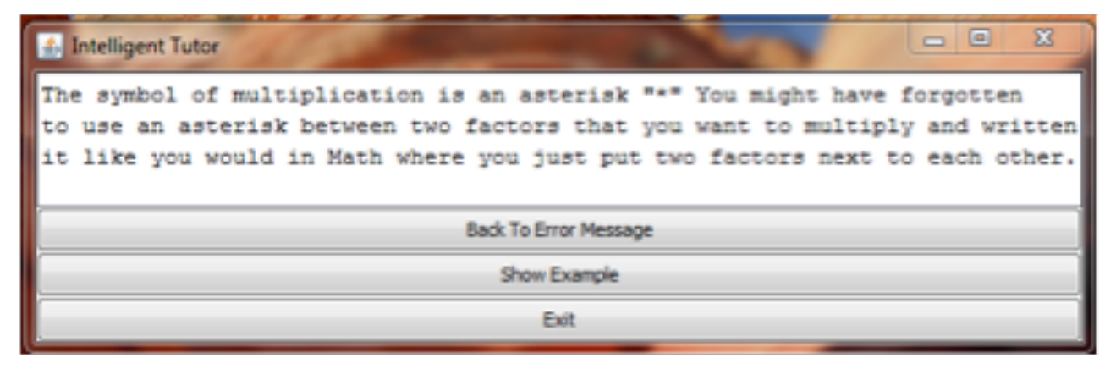
BraceExpectedErrorHandler
IncompatibleTypesErrorHandler
SemicolonExpectedErrorHandler
CannotFindSymbolErrorHandler
BracketExpectedErrorHandler
MissingReturnStatementHandler
IllegalStartOfStatementHandler

GRAPHICAL USER INTERFACE DEVELOPMENT

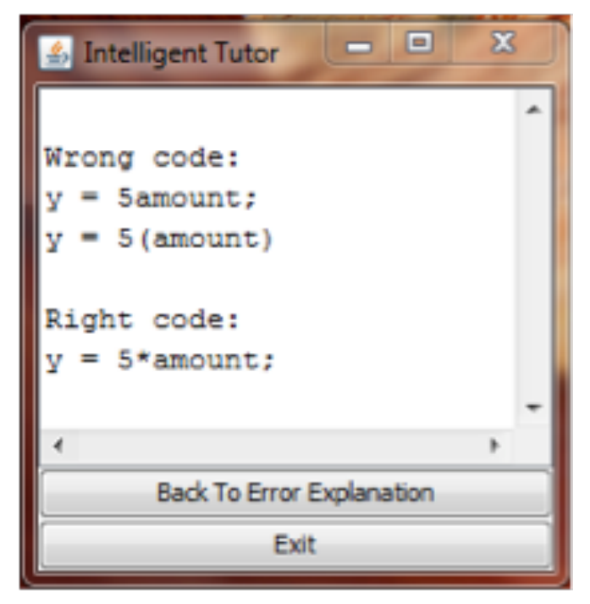
Most ITSs' user interfaces are more focused on the user than the performance. This "friendliness" of an ITS user interface is important in helping the student maintain the acquisition of knowledge by keeping his or her confidence and comfort in using the system on a high level [21]. Thus, the graphical user interface for this ITS will be modified to make it simpler and more userfriendly. The basis for defining 'user-friendly' will be first and foremost feedback from the novice programmers who have used the current ITS. Using this definition of user-friendliness, the current graphical user interface will be modified accordingly. Below are screenshots from the previous system developed by Go and Ligunas[9].


Didn't use'*' for multiplication operation

Layer 1 (Error Description)

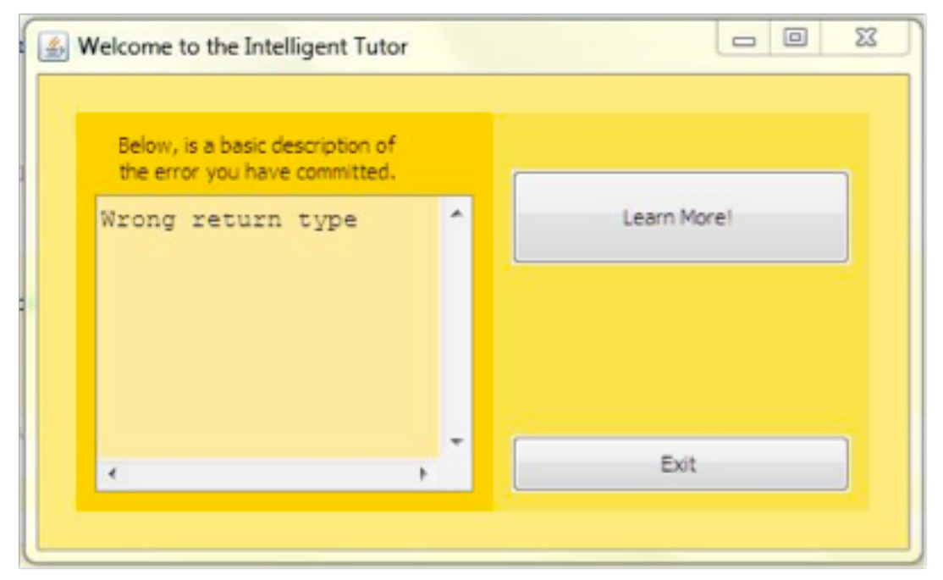

The symbol of multiplication is an asterisk "*" You might have forgotten to use an asterisk between two factors that you want to multiply and written it like you would in Math where you just put two factors next to each other.

Layer 2 (Suggested Solution)

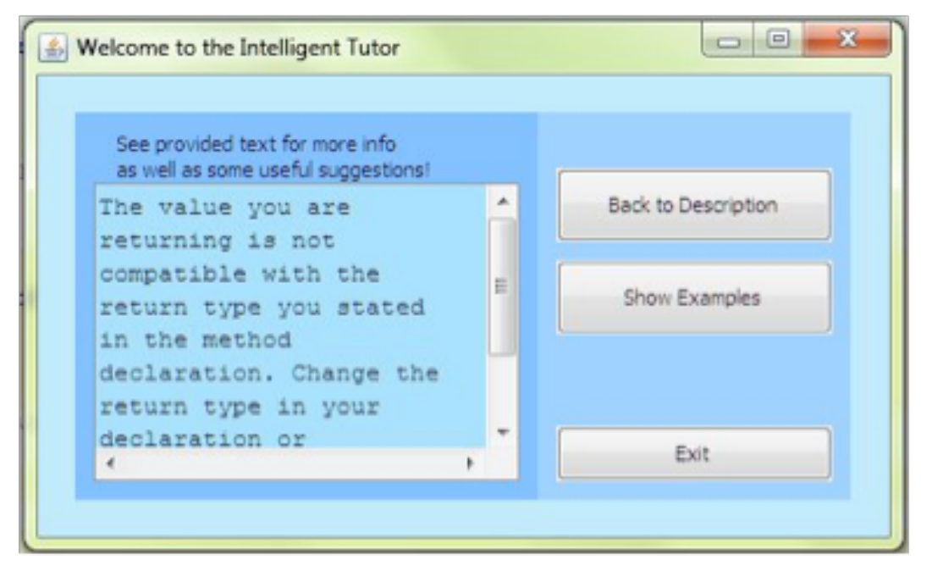

```
Wrong code:
y = 5amount;
y = 5(amount)

Right code:
y = 5*amount;
```

Layer 3 (Sample Wrong and Correct Code)

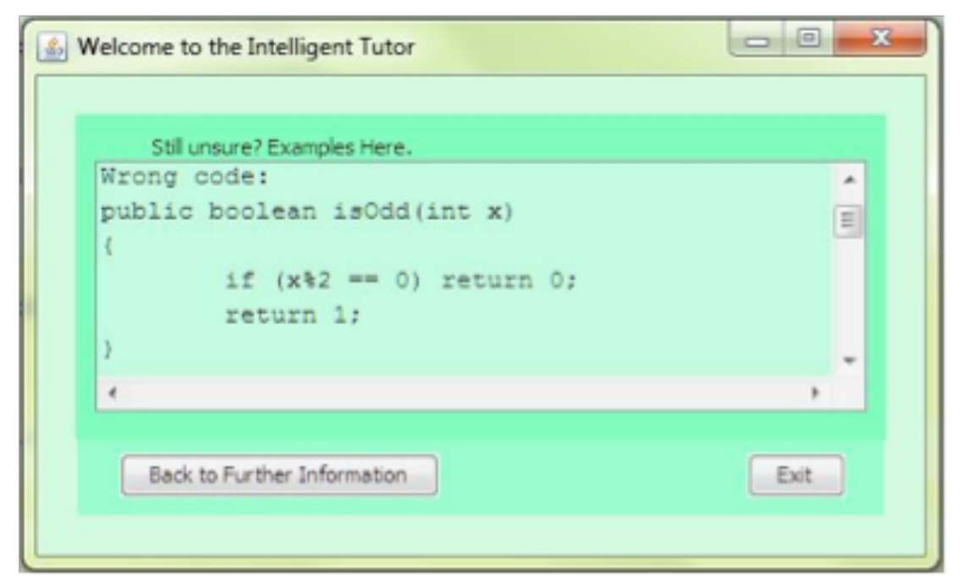
The group modified the interface of the system by adding bright colors to appeal to the eyes of the user.


Below, is a basic description of the error you have committed.
Wrong return type

Modified Layer 1 (Error Description)


See provided text for more info as well as some useful suggestions!
The value you are returning is not compatible with the return type you stated in the method declaration. Change the return type in your declaration or

Modified Layer 2 (Suggested Solution)


Still unsure? Examples Here.
```
Wrong code:
public boolean isOdd(int x)
{
        if (x%2 == 0) return 0;
        return 1;
}
```

Modified Layer 3 (Sample Wrong and Correct Code)

Aside from the appearance of the user interface, the group worked on improving the behaviour of the system, such as eliminating the persistence of the windows of the previous iteration of the ITS which may cause frustration for the users.

MODIFIED XML SCRIPTS

The group modified xml scripts developed by the previous group by correcting semantic errors. Below is a sample code of an error.

Brace Expected Error

```
<?xml version="1.0"?>
<BraceExpected>
    <error>
        <id>InvalidClassName</id>
        <a>Invalid class name.</a>
        <b>Valid class names are composed of Unicode letters, digits,
        currency symbols and underscores. They must not be separated by any spaces.
        By convention, class names are nouns with the first letter of each word capitalized.</b>
        <c>
Invalid class names:
    A)  Mobile Phone
    B)  Food|Drink
----------------------+0+----------------------
Valid class names:
    A)  MobilePhone
    B)  FoodAndDrink
        </c>
    </error>
    <error>
        <id>ClassAsType</id>
        <a>Class used as a type.</a>
        <b>The class itself is the type. Check that you did not use the word "class" as a type.</b>
        <c>
```

```
Wrong code:
    class GasStation = new GasStation();
----------------------+0+----------------------
Correct code:
    GasStation gs = new GasStation();
        </c>
    </error>
    <error>
        <id>ClassThrowsException</id>
        <a>Placed a throws Exception in class.</a>
        <b>You might have placed a 'throws Exception' in the class declaration.
        Place 'throws Exception' in method method declarations instead.</b>
        <c>
Wrong code:
    public class Driver throws Exception
    {
        public static void main(String args[])
        {
        }
    }
```

RESULTS

First Deployment

The first deployment consisted of nine (9) first year BS Computer Science students. The first deployment measured how well the ITS faired with human interaction. The students were tasked to work on a sample problem given to them using a copy of BlueJ that extends the ITS. The topics for evaluation afterwards consist of two parts: close-ended multiple choice feedback and open-ended written feedback. The close-ended part consists of thirteen (13) questions that revolve around design, content, and other aspects of the ITS that don't fall under the previous two categories. The open-ended part consists of six (6) questions in which the testers will respond with their experiences in using the ITS. Below is a tally of the results of the close-ended questions.

	SD	D	N	A	SA	Ave.
Design Feedback						
I thought the system is very easy to use.	0	0	0	3	6	4.67
It was simple.	0	0	0	3	6	4.67
The on-screen characters are easy to read.	0	0	0	2	7	4.78
I thought the system was too inconsistent.	1	4	3	1	0	2.44
The interface of this system is pleasant.	0	0	0	5	4	4.44
Content Feedback						
The ITS's eror identification is accurate.	0	0	2	4	3	4.11
The help messages are helpful.	0	0	1	3	5	4.44
The range of errors returned was sufficient	0	1	0	7	1	3.89
Others						
Bugs are easier to locate because of the ITS.	0	0	1	3	5	4.44
I recommend the ITS to novice programmers.	0	0	3	0	6	4.33
It will further improve my debugging skills.	0	0	2	4	3	4.11
I think I need the support of a technical person to be able to use this system.	2	3	2	0	2	2.67
I think that I would like to use this system.	0	0	2	4	3	4.11

DISCUSSION

Taking a look at the results of the first deployment, the improved interface satisfied the needs of the users. Though there is still room for improvement, the current system reached its objective. Adding new errors is out of our scope since another group will be dealing with it. Their output will help us widen the scope of error messages though the system that we created can stand on its own.

With the initial results gathered, the graphical user interface that will best suit the needs of novice programmers to help them in understanding and fixing the non-literal errors committed ws determined. With this, the current interface of the system with the default BlueJ background colors will be the best choice. The basic features of the system: nature of the error and examples about the error, will remain the same because it is where its helpfulness comes in.

As for the logs, further testing with the improved GUI is needed. The group will conduct tests to test it with the developed system