Francisco J. Almeida-Martínez
Jaime Urquiza-Fuentes
Antonio Pérez-Carrasco

Syntax error recovery visualization. Evaluation using VAST and Cup.

Number 2011-12

Serie de Informes Técnicos DLSI1-URJC
ISSN 1988-8074
Departamento de Lenguajes y Sistemas Informáticos I
Universidad Rey Juan Carlos

Index

Syntax error recovery visualization. Evaluation using VAST and Cup.

Francisco J. Almeida-Martínez, Jaime Urquiza-Fuentes and Antonio Pérez-Carrasco

Universidad Rey Juan Carlos
Departamento de Lenguajes y Sistemas Informáticos I
c/ Tulipn s/n, 28933 Madrid, Spain
francisco.almeida@urjc.es, jaime.urquiza@urjc.es,
antonio.perez.carrasco@urjc.es

Abstract. In this document we describe the evaluation of the syntax error recovery using VAST and Cup. In this case we performed a complete evaluation of the panic error recovery strategy. The results of the evaluation are satisfactory, so the numbers of deliveries and marks are higher in the treatment group than in the control one.

1 Introduction

In the preliminary evaluation of the syntax error recovery [1] we analyzed the problems that students found to understand the visualizations generated by VAST. In that evaluation students had to write certains input streams in order to make the syntax error recovery, which had been implemented into the parser, works in a specific way. The objective of this evaluation is different, so in this case, the students have to build the parser implementing the corresponding syntax error recovery. For this reason we only tested the panic syntax error recovery strategy.

The version of the tool used in this evaluation is the same that the one used in previous experiments [2]. However, we have finished the implementation of the insertion, deletion and panic mode error recovery. In previous evaluations we observed a special demotivation of students, so we decided to increment the incentive of this evaluation. Also we decided to reduce the number of questions about usability and quality.

2 Description of the evaluation

In this section we describe the evaluation. We refer to the participants, the experiment's design, the tasks performed during the session and the protocol.

2.1 Subjects

In this evaluation participated **19 students** of the *Language Processing* subject at the Rey Juan Carlos University during the 2010-2011 course. The participation was voluntary and based-incentive in a 2% over the final mark only if they passed the exam.

2.2 Experimental design

This evaluation was designed as an educational effectivenes study plus an usability and observational experiment. Students were divided in two groups, control and treatment. The control group used the tools JFlex-Cup and the treatment one VAST. In order to create the groups we used the marks of a pretest of knowledge. The students were divided using this mark but the assigment to control or treatment group was random. The pretest included questions about the syntax error recovery according to the Bloom's taxonomy [3]. In table 1 we show the results of the pretest of knowledge. As we can observe there does not exist any difference between both groups. The independent variable was the tool used; VAST in the treatment group and JFlex-Cup in control one. The dependent variables were the educational effectivenness using the differences between a pretest-postest and the students's opinions about three aspect related with the ease of use: general ease of use, learning support and quality of the tool. This evaluation lasted two hours (one session).

Level	Control	Treatment	Stats.
Knowledge (K)	0.04	0.05	U=335.00, p=0.73
Understanding (U)	0.02	0.03	U=345.50, p=0.89
Application (Ap)	0.03	0.03	U=332.00, p=0.48
Total	0.03	0.04	U=334.50, p=0.72

Table 1. Results of the pretest of knowledge according to the Bloom's taxonomy levels.

2.3 Tasks and protocol

The tasks performed during the evaluation had to be documented at the end of the session (see appendix 5) with text explanations and visualizations, using VAST in the treatment group and any other software in the control group. The tasks consisted in 2 exercises for parser design about the syntax error recovery. In table 2 we show the protocol followed in this evaluation.

Treatment group. This group had to use VAST to write the input streams asked in the exercises using the visual help offered by the tool. Also they had to use VAST in order to implement the corresponding syntax error recovery.

Control group. This group had to use the tools JFlex-Cup to solve the exercises without the support of any visualization tools. For each exercise the had to use a general porpuse editor and the corresponding generation tool. Also they had to performed the appropiate configuration of each tool.

Control	Treatment
Pretest of knowledge	
Syntax error recovery explanation	
JFlex-Cup session	VAST session
JFlex-Cup questionary	VAST questionary
Postest of knowledge	

Table 2. Protocol used in the evaluation

3 Results

During the evaluation the instructors observed how the students used the tools and the problems they found. The results are divided in instructors' observations, answers to questionaries and educational effectivennes.

3.1 Instructor's observations

During the evaluation in the treatment group we observed that students were used to work with VAST. After 40 minutes, all students had finished the first exercise. At the end of the evaluation, all students finished the exercises.

In the control group we observed that the most used visualization was the syntax rule used. One students required paper support to draw the syntax tree resulting of a syntax error. After 40 minutes, only one student had finished the first exercise. At the end of the session, none of the students had finished the exercises.

3.2 Answers to questionnaires

The opinion questionaries about VAST and JFlex-Cup were designed to make possible students give their opinion about the tools. We used a Likert scale with 5 values where the 1 was the lowest mark and 5 the highest. We performed an analysis of the marks obtained by each tool. In table 3 we show the results of the analysis.

Easy of use			
Aspect	**Control**	**Treatment**	**Stats.**
General ease of use	3.13	3.88	t(14)=-1.26, p=0.23
Learning support			
Panic error recovery	3.50	4.13	t(14)=-0.98, p=0.35
Technical quality			
General quality	3.13	4.13	t(14)=-1.61, p=0.13

Table 3. Usability-quality marks

3.3 Results of educational effectiveness

The results of educational effectiveness are divided in two parts. On the one hand, the differences between postest-pretest. On the other hand, the marks obtained in the exercises of the evaluation.

In table 4 we show the differences of the marks according to the Bloom's taxonomy levels.

Level	**Control**	**Treatment**	**Stats.**
Knowledge (K)	0.25	0.33	t(17)=-0.88, p=0.39
Understanding (U)	-0.06	-0.05	U=39.00, p=0.69
Application (Ap)	0.06	0.00	U=34.50, p=0.46
Total	0.05	0.04	t(17)=0.12, p=0.91

Table 4. Resutls of educational effectiveness according to the Bloom's taxonomy levels.

The solutions to the practices were analyzed in a global way (U=45.00, p=1.00) and independently by questions (U=36.00, p=0.73 for both questions).

4 Conclusions

The results of this evaluation are divided in two parts. On the one hand the results of educational effectivenes. On the other hand the students' opinion about the tool used.

According to the educational effectivenes, the results are divided in two parts: solutions to the practices and the differences between postest-pretest. The solutions to the practices do not show any significant difference neither in the global nor the individual analysis. However, we observed that the students in the treatment group were able to finish the exercises faster than students in control gruop.

5 Acknowledgment

This project is supported by project TIN2008- 04103/TSI of the Spanish Ministry of Science and Innovation.

A Exercise used for the pre-postest

1. What are the objetives of the syntax error recovery?
2. How does the insertion syntax error recovery strategy work? What is its limitation?
3. How does the deletion syntax error recovery strategy work? What is its limitation?
4. How does the panic error recovery strategy work? What is its limitation?
5. Given the following grammar:

   ```
   S::= T id (L) {B}
   T::= i|v|b
   L ::= T id, L | lambda
   B::= id = E B | lambda
   E ::= + FE | -FE | lambda
   F::= id | cte
   ```

   Using the notation **($Stack, Input Stream$)**, show cases of a non recursive LL (1) parser:

   (a) The recovery from an error using the insertion strategy.
   (b) The recovery from an error using the deletion strategy.
6. Using the grammar of the previous exercise:

   ```
   S::= T id (L) {B}
   T::= i|v|b
   L ::= T id, L | lambda
   B::= id = E B | lambda
   E ::= + FE | -FE | lambda
   F::= id | cte
   ```

   We have implemented a LL (1) parser with syntax error recovery using the insertion strategy. Given the input stream **id (cte + cte;** , draw the corresponding syntax tree indicating the input stream recognized, the syntax error and the recovery point.
7. Given the following grammar:

   ```
   S::= T id (L) {B}
   T::= i|v|b
   L ::= T id, L | lambda
   B::= id = E B | lambda
   E ::= E+F | E - F| lambda
   F::= id | cte
   ```

Show cases of a LR (1) parser specifying the input stream and the stack, before and after the execution of the syntax error recovery using the panic strategy using as synchronization points the *followings* of the antecedent which is being processed.

8. Using the grammar of the previous exercise:

```
S::= T id (L) {B}
T::= i|v|b
L ::= T id, L | lambda
B::= id = E B | lambda
E ::= E+F | E - F| lambda
F::= id | cte
```

We have implemented a LR (1) parser with syntax error recovery using the panic strategy, where the synchronization points are the FOL of the antecedent which is being processed. Using this input stream: **id + (cte –) - (id cte ;** . Draw the corresponding syntax tree indicating the part which is processed, the ignored part, the error detection point and the part of recovery.

B Exercises used in the evaluation

1. **Problem 1.** Given the following grammar:

```
E::= for (IDENT:=CTE_INT; IDENT OPR CTE_INT; IDENT INC ){ SENT } | E
for (IDENT:=CTE_INT; IDENT OPR CTE_INT; IDENT INC ){ SENT } SENT::=
SENT EXP | EXP; EXP ::= eval  (EXP2)
    | ident :=EXP
    | EXP2;
EXP2 ::= EXP2 and L
    | EXP2 or L
    | L;
L::= L nand T
    | L xor T
    | T ;
T::= IDENT
    | CTE_INT;
OP ::=  ==| >|<; INC::= ++|--;
```

It has been implemented in the files *problem.flex* and *problem.cup*.

(a) Using the panic strategy, implement the syntax error recovery for an error because the user forgets the first ";" in the condition inside of the *for* sentence. The parser has to ignored the minimun number of tokens as possible.

(b) Using the panic strategy, implement the syntax error recovery for an error because the user forgets the first ";" in the condition inside of the *for* sentence. The parser has to ignored the maximun number of tokens as possible.

(c) Using the panic strategy, implement the syntax error recovery for an error due to a wrong expression in EXP2 inside the *eval* body. The parser has to ignore the minimun number of tokens of the input stream.

References

1. Francisco J. Almeida-Martínez and Jaime Urquiza-Fuente. Preliminary evaluation of the syntax error recovery visualization using VAST. Technical Report 2011-08, Universidad Rey Juan Carlos, 2011.
2. Francisco J. Almeida-Martínez and Jaime Urquiza-Fuente. Building LR(1) grammars and parsers using VAST and Cup. Technical Report 2011-11, Universidad Rey Juan Carlos, 2011.
3. B. Bloom, E. Furst, W. Hill, and D.R. Krathwohl. *Taxonomy of Educational Objetives: Handbook I, The Cognitive Domain.* Addison-Wesley, 1959.