Francisco J. Almeida-Martínez
Jaime Urquiza-Fuentes

VAST. Evaluation of the execution-visualization functional integration.

Number 2011-07

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

Index

VAST. Evaluation of the execution-visualization functional integration

Francisco J. Almeida-Martínez, Jaime Urquiza-Fuentes

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

Abstract. This document describe an observational and usability evaluation of VAST in the course 2008-2009. This evaluation was performed after implementing the first functional integration called *execution-visualization*. The results of this evaluation show that students like VAST but it is necessary to add new features as the second functional integration, *annotation-generation-compilation*.

1 Introduction

From the results of previous evaluations [1], we plan the necessity to adapt the process of visualizations generation to the real process of parser generation. Due to this, we planned two functionality integrations. One of this integration is the *execution-visualization*. This functionality was added to VAST, so the user can edit the input stream and run the parser from the own user interface. Thanks to this, the user can generate new visualizations without effort and observe how the parser behave for different input streams. The objective of this evaluation is observe the usability of this new integration.

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, the protocol and the results.

2.1 Subjects

In this evaluation participated **5 students** of the subject *Compilers and Interpreters* of the Rey Juan Carlos University in the 2008-2009 course. The evaluation was incentive-based in a 3% over the final mark only if the passed the exam. The gender distribution was not balanced, so only the 20% (1/5) of the students were women.

2.2 Experimental design and tasks

This evaluation was designed as an usability experiment plus an observational study. In this evaluation the students were divided in two groups. For this task we used a class exercise as a pretest of knowledge (see appendix 5).

The tasks performed during the evaluation consisted in two exercises (see appendix 5). From the lexical and syntax specifications given, the students had to annotate, generate the parser and executed it, using the new functional integration, *execution-visualization* of VAST. When the groups were created, the students had to annotate the lexical or the syntax specification. The objective of this task was to take note of the difficulties of each part.

The dependent variables were the students' opinion about four different aspects: the ease of use, quality, satisfaction and learning help. The experiment lasted 2 hours (one class). During this time the students had to use VAST's documentation. At the end of the session we asked the students for answering an opinion questionnaire about the tool, including the new integration.

2.3 Protocol

Two weeks before the evaluation we performed a pretest of knowledge, which was done by only two students. For this reason we did a random division of the participants being impossible to balance both groups. One week before the evaluation we presented VAST and how to work with the tool. This lasted about 30 minutes, during the rest of the session students worked on the class exercises. In the table 1 we show the protocol followed in this evaluation.

Group 1	**Group 2**
Pretest of knowledge	
Introduction to VAST	
Lexical annotation 1	Syntax annotation 1
Syntax annotation 2	Lexical annotation 2
Answers to questionnaries	

Table 1. Protocol followed in this evaluation

3 Results

In this section we describe the results of the evaluation. During the experiment the instructor observed how the students worked with the tool. The results are divided in two categories: instructor's observation and answers' to questionnaires.

3.1 Instructor's observations

During the evaluation we observed how the students performed some of the tasks of the test. Some students did changes in the visual representation of the syntax tree using the configuration tool of VAST. Others were more interested on the changes of the user's interface (distribution of views). In general, students preferred the used of the floating view instead of using the horizontal/vertical distribution. It was interesting that students made a high use of the parser's stack and the automatic animation of the syntax tree building process.

Furthermore, we observed that students required the *annotation-generation-compilation* integration.

3.2 Answers to questionnaires

The results of the questionnaires are focus in the values of the dependent variables: ease to use, general quality of the tool, student satisfaction and learning support.

Aspect	**% of answers between 4/5**
Ease to use	100%(5/5)
Quality of the tool	100% (5/5)
Student satisfaction	80%(4/5)
Learning support	80%(4/5)

Table 2. Summary of questionnaires answers

In table 2 we show a summary of the results obtained in the opinion questionnaires filled after the evaluation by students.

In the questionnaire we also included open questions in order to make possible students to express their opinion easily. As positive aspect students pointed out the visualization of the syntax tree, its building process, the ease of use and the user interface. As posible improvements they mentioned the interaction with the syntax tree, the automatic lexical/syntax annotation and the exportation of animations.

4 Conclusions

The results obtained show that the development of VAST is centered on the students.

As it can be observed in table 2 most of the answers or sometimes all of them, have a high mark. According to the ease of use, VAST obtained a high mark and students did not any negative comment. In relation with the quality of the tool, results were also satisfactory. The general satisfaction of the students obtained a worse mark due to the annotation process. Two students describe this process

as hard and monotonous. Because of this the general satisfaction of VASTapi is lower. Finally, the learning improvement also receive a worse mark. One student gave a 3 to the use of the stack.

As result of this evaluation we plan to implement the another integration *annotation-generation-compilation*.

5 Acknowledgment

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

A Exercise used for the pretest

Exercise:

WWW stands for *World Wide Web*, a global net based on the architecture to transfer hypertext files between machines connected to the internet. To make posible the localization of a hypertext file on the internet it has been defined a language which sentences are called URL (Uniform Resource Locator). The URL allows to specify the communication protocol, the machine and the file. For example, the following URL *ftp://apolo.uma.es/dist/doc/drol.ps* shows that we want to obtain the file */dist/doc/drol.ps* from the machine *apolo.uma.es* using the protocol *ftp*. For this exercise we only consider that the WWW only admits the following protocolos: *http*, *ftp* and *gopher*. The name of the machine can be written in two ways:

- In symbol notation, as in the previous example, using the machine's name and (optionally) the domains, separated by the point character, for example: *apolo* or *apolo.uma.es*.
- Using decimal notation, specifying the four numbers of the internet direction, for example: 150.214.58.55.

The file's name of the URL address is any correct name of a valid file in UNIX including the complete path (using only alphanumerics characters, the point and the slash to separate directories), for example: *index.html* or */dist/doc/drol.ps*. In a URL reference the protocol is optional, so by default it asumes the *http* protocol. Also the file's name is optional, so it uses by default *index.html*. For example, the following address are valid and equivalent: *apolo.uma.es*, 150.214.58.55, *apolo.uma.es/index.html* and *http://apolo.uma.es*.

Using the tools JFlex and Cup, build a parser to recognize the URL address. A possible grammar could be:

$S ::= URI\ S | URI$

$URI ::= PROTOCOLO : //NOMBRE\_MAQUINA\ URI2 | NOMBRE\_MAQUINA\ URI2$

$URI2 ::= DIRECCION\_FICHERO\ | /NOMBRE\_FICHERO | \lambda$

$NOMBRE\_MAQUINA ::= DOMINIO | DIRECCION\_IP$

$PROTOCOLO ::= FTP|HTTP|GOPHER$
$DIRECCION\_IP ::= DIGITO\ .\ DIGITO\ .\ DIGITO\ .\ DIGITO$
$DOMINIO ::= IDENT|IDENT\ .\ DOMINIO$
$DIRECCION\_FICHERO ::= /IDENT/NOMBRE\_FICHERO|/IDENT\ DIRECCION\_FICHERO$
$NOMBRE\_FICHERO ::= IDENT\ .\ IDENT$

B Exercises for the evaluation

1. Exercise 1: Given the following grammar to declare procedures/functions:
   $S ::= TYPE\ ident\ (LISTPARAM)\ SENTLIST$
   $TYPE ::= boolean\ |integer\ |float$
   $LISTPARAM ::= TYPE\ ident|TYPE\ LISTPARAM,\ ident$
   $SENTLIST ::= SENT|SENTLIST;\ SENT$
   $SENT ::= ident\ =\ EXP;$
   $EXP ::= EXP\ +\ T|T$
   $T ::= T\ *\ F|F$
   $F ::= ident\ |(EXP)\ |cte$

   (a) Build an animation using VAST. The animation must show the whole declaration of a function with N parameters and at least one assignment instruction.
   (b) Modify the input stream to include four assignment instructions.
2. Exercise 2: Given the following grammar to declare SQL queries:
   $S ::= LQUERY$
   $LQUERY ::= LQUERY\ ;QUERY|QUERY$
   $QUERY ::= select\ LID\ in\ LISTTABLAS\ where\ LISTCOND$
   $LID ::= LID,\ ident|ident$
   $LISTTABLAS ::= LISTTABLAS\ ident|ident$
   $LISTCOND ::= F\ OP\ LISTCOND|F\ OP\ F$
   $OP ::= and|or|nand|xor$
   $F ::= ident|cte$

   (a) Build an animation using VAST. The animation must show the whole declaration at least one SQL query with at least two conditions.
   (b) Modify the input stream to include two SQL queries with at least two conditions each one.

References

1. Francisco J. Almeida-Martínez and Jaime Urquiza-Fuente. Teaching LL(1) parsers with VAST- An Usability Evaluation-. Technical Report 2009-01, Universidad Rey Juan Carlos, 2009.