

Companion Documentation for VL-HCC 2017

Paper by Bastos, Afonso and de Souza

Paper Citation: Bastos, J. A. M. D. ; Afonso, L. M. ; de Souza, C. S. (2017) Metacommunication Between Programmers Through an Application Programming Interface. In: **IEEE Symposium on Visual Languages and Human-Centric Computing (VL-HCC 2017) - Proceedings**. IEEE Symposium on Visual Languages and Human-Centric Computing (VL-HCC 2017). October 11-14, 2017 – Raleigh, NC. Piscataway: IEEE. p. 213-222.

Technical Report Author: João A. D. M. Bastos

Introduction

This document complements information in our paper published in VL/HCC 2017, the IEEE Symposium on Visual Languages and Human-Centric Computing (see citation above). It summarizes the steps taken in our study and provides a more detailed view of the data we have collected in the interviews conducted with seven experienced programmers. However, since all the interviews were in Portuguese, what we present here is a translation of selected items, not the item itself.

We start by describing the procedures for collecting the data from participants in the interview, and then we present a more detailed view of the collected data, which due to lack of space in the paper may need further information than we have been able to provide there.

Data collection procedures: Interview with Participants

We interviewed seven professional programmers. We used guiding questions with to explore their experience and feelings regarding APIs in general and the date and time APIs in particular. In addition to the questions, we also used a quiz, where we asked the participants to fill in a table with the expected results for two operations: “add a month to January 31, 2016” and “create the date February 31, 2016”. After the quiz, we had follow-up questions about the reasons why the programmers thought the results would be what they indicated in the corresponding table cells. The interview script (translated from the original Portuguese) is as follows.

Initial Questions

01 - In your opinion, what aspects are desirable in an API? And which ones are undesirable?

02 - In your experience as a programmer, which of the aspects you just mentioned have you come across? What happened on those occasions?

Both questions were open to subsequent exploratory questions if the interviewer felt it would lead to richer information.

The Quiz

In the quiz, participants were then invited to fill in a table form with the expected outcome for each operation, in each one of nine different. In addition, they should also indicate their experience on a scale from 1 to 5, where 1 would be “I know very little or I do not know the API” and 5 would be “I have full knowledge of the API” for each one of the APIs investigated. The blank table received by participants looked as shown below (contents were originally in Portuguese):

	Add a month to date January 31, 2016	Create the date February 31, 2016	On a scale from 1 (lowest value) to 5 (highest value), indicate:	
			Your familiarity with LP	Your familiarity with API
Java 7				
Java 8				
Python				
Php				
SQL				
Lua				
C#				
JavaScript				
Moment.JS				

Questions for after the quiz

01 - I see different results (for different APIs and operations) in your table. What is different about *the APIs* that led you to believe they should yield different results for the same operations?

OR

01 - I see that the same results (for different APIs and operations) in your table. What is the common trait to all these APIs that led you to believe that they should yield the same results for all operations?

02 - May your prior knowledge of (certain) **programming languages** have influenced your answers? Can you think of any aspect or way of functioning of some particular programming language that may have influenced you (while filling out the table slots)?

03 - May your prior knowledge of (certain) **API** have influenced your answers? Regardless of whether you know the specific part that we are talking about, do you think that some behaviors in other parts of some API may have influenced your answers?

Question Confronting Actual Results

01 - Looking at the results that the APIs actually produce, can you comment on the results that you thought they would produce?

02 - What do you think differentiates these APIs from one another? Do think these differences can explain why they produce different results?

Collected data

We separate the presentation of the collected data into two parts. One is a quantitative compilation of the participants' answers for the quiz. The other is the input for qualitative analysis, where we categorize and interpret the participants' statements.

In table below, we present the quantitative compilation. The first column indicates the names of the programming language packages in which the data and time API is included. The second column indicates all the answers proposed by participants for the operation "add a month to date January 31, 2016" and the third column the results for the operation "create the date February 31, 2016". The first value of each cell, in boldface, is the value actually produced by the corresponding API. The total number of coincidental answers is shown in parentheses.

	Add a month to date January 31, 2016	Create the date February 31, 2016
Java 7	29-FEB-2016 (2) 31-FEB-2016 (1) 01-MAR-2016 (1) 03-MAR-2016 (1) 02-MAR-2016 (1) ERROR (1)	02-MAR-2016 (1) ERROR (5) 03-MAR-2016 (1)
Java 8	29-FEB-2016 (2) 01-MAR-2016 (2) 03-MAR-2016 (1) 02-MAR-2016 (1) ERROR (1)	ERROR (7)
Python	29-FEB-2016 (1) 01-MAR-2016 (2) 28-FEB-2016 (1) 02-MAR-2016 (2) 31-FEB-2016 (1)	ERROR (7)
Php	02-MAR-2016 (2) 31-FEB-2016 (2) 01-MAR-2016 (1) 28-FEB-2016 (1) ERROR (1)	02-MAR-2016 (0) 31-FEB-2016 (2) 01-MAR-2016 (1) ERROR (4)
SQL (Postgres)	29-FEB-2016 (1) 01-MAR-2016 (2) 28-FEB-2016 (1) 02-MAR-2016 (2) ERROR (1)	ERROR (5) 31-FEB-2016 (2)
Lua	02-MAR-2016 (1) 31-FEB-2016 (1) 01-MAR-2016 (1) 28-FEB-2016 (1) 29-FEB-2016 (2) ERROR (1)	02-MAR-2016 (0) 31-FEB-2016 (1) 01-MAR-2016 (1) ERROR (4) 29-FEB-2016 (1)
C#	29-FEB-2016 (2) 01-MAR-2016 (2) 28-FEB-2016 (1) 02-MAR-2016 (1) ERROR (1)	ERROR (7)
JavaScript	02-MAR-2016 (2) 31-FEB-2016 (1) 01-MAR-2016 (1) 28-FEB-2016 (1) 29-FEB-2016 (1) ERROR (1)	02-MAR-2016 (0) 31-FEB-2016 (1) ERROR (5) 29-FEB-2016 (1)
Moment.JS	29-FEB-2016 (1) 31-FEB-2016 (1) 01-MAR-2016 (1) 28-FEB-2016 (1) 02-MAR-2016 (2) ERROR (1)	ERROR (4) 31-FEB-2016 (1) 01-MAR-2016 (1) 29-FEB-2016 (1)

During the interviews, we recorded all participants' responses for further analysis. In the analysis process, we categorized the participants' statements into groups (*codes*) that we found relevant for our research. Below is the list of categories and excerpts of translated testimonials that have been assigned to them.

Ease of setup

"So, [with] some APIs I need to set environment variables and these things. This is difficult for me. I don't like it. The less I have to configure, the better environment." - Participant 1

"Some APIs that have several stages of "edification". [But] When you work with APIs that return only JSON, it's much simpler" - Participant 6

"It has to be a quick API. APIs in Xml bother me. You cannot do it fast, it is not practical. [When it comes to] using them, you have to be doing lots of treatment. I prefer APIs with JSON." - Participant 7

Convenient to use

"Simple in the sense that you do not have to send out a very complex structure of information. Makes it quick to do [what you want]." - Participant 6

"Sometimes you have a similar set of functions and you are getting confused. When to use one, when to use another one ... The guy who's doing the API should think about it. Make it clear and take you by the hand, so you need to read as little documentation as possible. Ideally you should just glance at it and know what it works for." - Participant 3

Documentation Needed

"First you have to have a good manual, so I understand how to use that API. Good documentation" - Participant 1

"To begin with you have to have good documentation. Good examples of use. Documentation of API methods, you must have clear examples and the simplest possible" - Participant 2

"I think the most interesting thing to have in the API is the documentation. A very consistent documentation and have clear examples." - Participant 4

"I like it when it has documentation and it helps to be less bureaucratic" - Participant 6

"In the background I would like to have a manual detailing the entire API" - Participant 5

"Having an example that is easy to understand and use by myself might even [let us do without] documentation." - Participant 5

Third Parties Examples

“Another thing I [think of] is comments from third parties, other people who needed to use that API” - Participant 1

“The first step I take when I'm going to study an API is to look at some Wiki, a blog that explains the general steps of what's good for me and quick examples of use and what I get with that API” - Participant 2

“When I get third party software, what I'm looking for right now is to get small, ready-made examples of how the library is used to try to reproduce them and do what I need in the fastest possible way.” - Participant 5

Consistency

“Sometimes the API is poorly documented. Some methods say they do something but they do something else. Sometimes it is an error in the implementation and sometimes it is an error in the documentation.” - Participant 2

“That the API be consistent with naming issues, return patterns, and so on. The date API for example. For months he works with arrays, but for days he works with numbers. So, you can never know which is the starting value of those, if zero or if one.” - Participant 4

“Inconsistency in Java7. Since I knew the sum would be [what it is], I thought that the creation of the date would work the same way, but it didn't.” - Participant 4