I_mean_something_to_somebody (part 2 of 2)

Experiment performed at the 2003 ACCU Conference

First published in CVu vol. 16 no. 2

Derek M. Jones
derek@knosof.co.uk

1 Introduction

This is the second of a two part article describing an experiment carried out during the 2003 ACCU conference. The first part was published in a previous issue of C Vu ref??? and discussed the background to the experiment and some of the applicable characteristics of the subjects taking part; this one, the second, discusses the results of the experiment.

The aim of this experiment was to measure one particular aspect of software developers behavior when assigning meaning to identifier names. This aspect was the extent to which knowledge of the application domain of the source code containing an identifier affects the meaning developers assign to that identifier name.

Software developers are constantly exhorted to use *meaningful* identifier names. However, there have not been any published studies investigating the kinds of information readers extract from identifier names or of any benefits the availability of this information might provide to readers. Reading source code whose identifier names are based on a human language the reader does not speak provides a vivid example of the, often unappreciated, benefit that identifier names can provide to readers (when these names are based on a human language spoken by the reader).

```
if ( pParametreFichier != (FILE*) NULL )
   {
   memset( &Enregistrement.CodeInterne1 , ’\0’ ,
            sizeof( Enregistrement.CodeInterne1 ) );
   memset( &Enregistrement.BlocPrimaireNumerique , ’\0’ ,
            sizeof( Enregistrement.BlocPrimaireNumerique ) );
   while (!ExcTrouve)
      ...
```

2 Background

Words are used both to communicate with other people and for internal thought processes. The culture we are born into provides us with a predefined set of words and a network of meanings associated with them. The use of words in their spoken form, to communicate with other people, has a cost that speakers attempt to minimise, by using them in a way that is consistent with the meaning they believe their listeners will assign to them. A lifetime of realtime feedback, from the people spoken to, enables users of a language to build a detailed collection of beliefs on the meanings assigned to words by both people in general and some specialist groups of people (e.g., software engineers).

When speaking it is expected that not only will listeners make an effort to comprehend the speakers thought processes, but that speakers will make an effort to ensure that what they are saying is comprehensible to their listeners. When writing text people must make use of their experience with the spoken form to help ensure that readers will assign a meaning to the words that is consistent with that intended. However, there is no realtime feedback between writer and reader[1] and experience shows that readers often have to invest significantly more effort to assign a coherent meaning to what they read, compared to the effort needed while listening during a spoken conversation.

Software developers are not usually told which identifiers they should use in a given context and are rarely given rules for creating new identifier names from existing ones.[2]

2.1 Selecting identifiers

Experience shows that many developers believe that the names they select for identifiers are *obvious*, *self-evident*, or *natural*. Studies of peoples performance in creating names for objects suggests that this belief is false.[?, ?, ?] When asked to provide names for various kinds of entities people have been found to select a wide variety of different names, showing that there is nothing *obvious* about the choice of a name.

[1] *Talking* via text messaging is not discussed here.

[2] The high cost of having database fields representing the same data item, e.g., a persons first name, but with different names, e.g., `first_name` or `given_name` or `christian_name`, across multiple databases has caused some organizations to plan to start mandating the use of specific names to denote specific data items.[?]

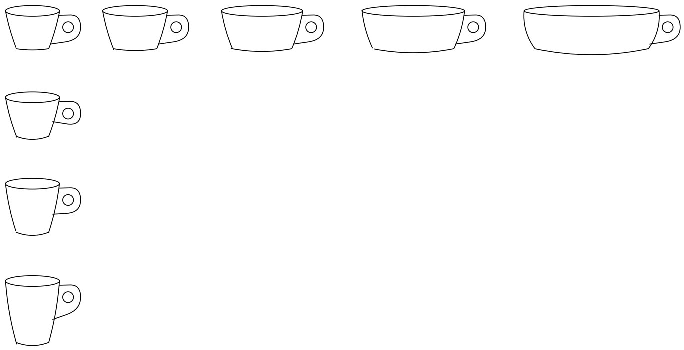

Figure 1: Cup and bowl like objects of various widths (ratios 1.2, 1.5, 1.9, and 2.5) and heights (ratios 1.2, 1.5, 1.9, and 2.4). From Labov.[?]

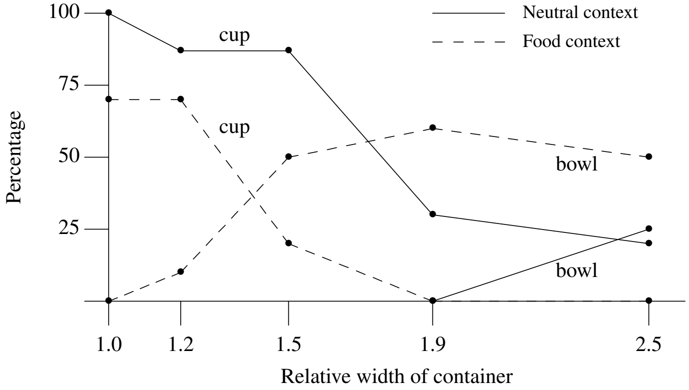

Figure 2: The percentage of subjects who selected the term *cup* or *bowl* to describe the object they were shown (the paper did not explain why the figures do not sum to 100%). From Labov.[?]

One naming study[?, ?] described operations (e.g., hypothetical text editing commands, categories in *Swap 'n Sale* classified ads, keywords for recipes) to subjects, who were not domain experts, and asked them to suggest a name for each operation. The results showed that the name selected by one subject was, on average, different from the name selected by 80-90% of the other subjects (one experiment included subjects who were domain experts and the results for those subjects were also consistent with this performance). The number of occurrences of different names chosen tended to follow an inverse law with a few words occurring frequently and most only rarely.

Various factors have been found to influence the selection of what is believed to be the appropriate word in a given context. A study by Labov[**?**] showed subjects pictures of individual items that could be classified as either cups or bowls, Figure 1. These items were presented in one of two contexts; a neutral context in which the pictures were simply presented and a food context (subjects were asked to think of the items as being filled with mashed potatoes).

The results showed, Figure 2, that as the width of the item seen was increased, an increasing number of subjects classified it as a bowl. By introducing a food context subject responses are shifted towards classifying the item as a bowl at narrower widths.

2.2 Recognizing words

Human languages have a relatively fixed set of letter sequences that are acknowledge by speakers of a language as being *root words* (this glosses over the heated discussions that sometimes occur over what letter sequences should be treated as root words). Additional words can be derived from these words using languages specific rules (e.g., "*write*" ⇒ "*writes*", "*writing*", "*written*"; "*writer*" could be treated as either a derived or a root word).

Identifiers sometimes contain more than one word. In this case readers need to either use their knowledge of existing words, to subdivide an identifiers character sequences, or use deduction based on common naming conventions to extract words (e.g., `IsHot` is likely to be interpreted as the phrase *is hot*, rather than *I shot*).

Identifiers often have the form of one or more abbreviated words. A study by Ehrenreich and Porcu[?] found that readers performance in reconstructing the original word, from an abbreviated form, was significantly better when they knew the rules used to create the abbreviation (81-92% correct), compared to when the abbreviations rules were not known (at best 62% after six exposures to the letter sequences). Given that this experiment was not intended to measure subjects abbreviation to word reconstruction performance, no rarely occurring abbreviations were used.

2.3 Studies of meaning assignment

While there have been no other published studies of how people assign a meaning to identifiers there have been a few studies of a similar nature for words.

A study by Nickerson and Cartwright[?] asked subjects to write down as many different meanings of a word (presented one at a time, in written form, for 30 seconds). Combining the results from all subjects showed that words were often given over 6 and sometimes as many as 20 different meanings. The majority of the responses for a given word were usually contained within one or two meanings.

Word association is an activity that has some similarities to providing a meaning for a word. Studies of word association give subjects a word and ask them to write down the first meaningfully related word that comes to mind. (e.g., *doctor* ⇒ *nurse*).

The results of these studies [3] have found that there is rarely a single answer, wide range of responses given, words given by subjects do not always overlap those of other subjects

A subjects age has also been found to be a factor in word association performance. A study by Hirsh and Tree[?] compared the responses of young (21-30) and older (66-81) adults to 90 stimulus words. The results showed that the same word was produced as the most popular response, for a given age group, in 36 out of 90 cases (when the top three responses were considered the overlap between groups was 57%). They also found that the younger group produced a wider range of responses, and that members of the older group were much more likely to select the most popular response for their group (40%, against 20% for the younger group).

3 Experimental setup

The experiment was run during two, on different days, 30 minute sessions of the 2003 ACCU conference held in Oxford, UK. Subjects were given a brief introduction to the experiment, during which they filled out background information about themselves. They then spent 15 minutes working on the identifier list. All subjects volunteered their time and were anonymous.

The first part of this paper describes background of subjects and how this information was collected.

Almost any sequence of characters could serve as an identifier. However, the initial list of identifiers considered for use in the experiment were obtained by extracting all identifiers that were common to the source code of a variety of programs. These programs were the Linux kernel, the game DOOM, GCC (the GNU compiler collection), Netscape internet browser, Postgresql database, AFS (Advanced File System) from IBM, and OpenMotif from the OpenGroup. It was hoped that usage in a wide variety of programs was an indication that an identifier had a significant meaning to a large number of developers. This method also removes experimenter bias from the choice of identifier names (but not from the choice of programs to consider).

The initial list was refined by removing those identifiers that were the names of standard library functions (these might be recognized as such and their library meaning given as a response), or contained rarely

[3] The University of South Florida word association norms[?] lists nearly three-quarters of a million responses to 5,019 stimulus words produced by 6,000 participants.

occurring abbreviations, or contained a single character. The resulting list of identifiers was randomized and printed one per line on A4 sheets of paper.

All subjects from both groups saw an identical list of identifiers. However, one group was told that the identifiers came from a multiplayer game, while the other that they came from the Linux kernel. The instructions given were:

> The following pages contain identifiers that have been extracted from the source of {a very large multiplayer game program}/{the Linux kernel}. For each identifier:
>
> 1. when you first see the identifier, write down any ideas that pop into your head about what it might represent,
> 2. briefly (5-10 seconds is sufficient) think about what the identifier might represent. Write any new ideas you have on a separate line.

3.1 Threats to validity

There are a number of reasons why the responses given in this experiment might not be valid in a source code comprehension context. These include:

- developers are not usually asked to provide the kind of information that they were asked to provide in this experiment. It is possible that the subjects were unsure of the responses expected of them, or misinterpreted the instruction they were given,
- identifiers invariably exist within a context when they are read in source code. For instance, there are other identifiers (e.g., the name of the function in which an identifier is referenced) whose names often provide a subcontext,
- providing a possible meaning for an identifier requires a lot of intellectual effort. It is unusual for developers to be asked to provide a meaning to so many identifiers over such a relatively short period of time. Over the period of the experiment fatigue may have caused subjects performance to decline, because of the high cognitive work load,

A few of the subjects had a different cultural background from the majority of the subjects (i.e., they were not British). It is possible that these subjects made use of different cultural conventions when assigning meaning to identifiers. For instance, in the US politicians *run* for office, while in Spain and France they *walk*, and in Britain they *stand* for office.

It is possible that on the first day I failed to point out, during the introduction, that the identifiers were extracted from a multiplayer game (I did point out that the identifiers came from the Linux kernel on the second day). This information is given in the instructions, but it is possible that subjects did not read the sentence containing this information.

4 Results

The 45 subjects produced a total of 1662 responses (34.8% Linux, 65.2% game), and 74 different words were responded to. There were 179 responses (45 different words) where the subject had written “*none*” (or a question mark, or a dash). The identifiers were printed on both sides of the page and some subjects only gave responses for identifiers appearing on the odd numbered pages. In this case the identifiers appearing on the even numbered pages were not counted as “*none*”.

Each subjects response for each identifier needed to be classified. The following process was intended to ensure that the person doing the classification (your author) was not influenced by information about the subject who gave the response. (i.e., whether the subject belonged to the Linux or games group, and which responses were given by the same subject). Every response was automatically assigned a random number and the resulting list of identifier/response pairs was sorted. This list of randomised responses was the one used for classification.

Certain words and phrases occurred several times in the responses and were assumed to imply a game context, but not a Linux context. These included: *player*, *game*, *skill level*, and *shoot*. While some words appear to have an obvious games meaning (i.e., *kill*), if it was possible that they also had a Linux meaning they were not classified as being games related.

Words and phrases that might be claimed to be a strong indication of a Linux context (e.g., *Linux*, *operating system*) rarely occurred in the responses. Much of the functionality provided by an operating system (e.g., Linux) might reasonably also be expected to be provided internally within a game. For instance, *virtual memory* refers to a memory management mechanism used by both operating systems and games (which, for efficiency reasons, might swap unneeded game information out of fast memory. This overlap in functionality, which many subjects are likely to be aware of, makes it difficult to reliably classify any responses as belonging to a Linux context.

A games context was assigned to 134 responses (12.4% of responses made by games subjects) scattered over 33 different words. A Linux context was assigned to 10 responses (1.2% of responses made by Linux subjects) scattered over 6 different words.

The forms of the meanings given were such that it was rarely possible to definitely specify which group a response belonged to. For instance, for the identifier `blue_pos` many subjects gave a response of the form *position of some blue thing*. In itself this response is not sufficient to be able to assign a Linux or game context. Additional information such as *index into array* could apply in either context. While use of the word *player* would suggest a game context.

In many cases the responses described a possible role that the identifier might fill, e.g., flag, or counter. While in other cases subjects simply expanded an identifier to a non-abbreviated form, e.g., gave *page number* as the response to `pagenum`.

Table 1: The five most common responses for identifiers having more than 20 responses (*most* indicates that most responses had this form).

Identifier	Number of Responses	Response (number)	Response (number)	Response (number)	Response (number)	Response (number)
accurate	27	flag (12)	none (6)	numeric value (4)	game (3)	
answer	29	input value (9)	result (4)	none (4)	string value (2)	game (1)
blue_pos	42	game (15)	none (14)	position of (10)		
body	30	none (7)	game (7)	code (7)	html (3)	
children	19	tree structure (6)	processes (3)	OO (3)	counter (2)	none (1)
cur_mode	40	cursor (4)	current mode (24)	none (2)	game (2)	linux (1)
def	19	definition (6)	define (6)	none (2)	language preprocessor (2)	
digest	44	cryptography (12)	summary (9)	eat (8)	none (6)	game (3)
disconnected	28	flag (most)	not connected (1)	game (1)		
driver	38	device driver (15)	game (5)	none (4)		
drop	44	delete/discard (11)	game (8)	none (4)	connection (4)	
event_mask	21	bit map/mask (all)				
force	22	physical force (8)	none (4)	flag (4)		
fraction	23	mathematical (16)	ration (2)	none (2)		
fragstotal	32	total fragments (12)	game (10)	memory fragments (2)	’frags’ (2)	none (1)
inactive	24	flag (most)	none (1)	game (1)		
inc	33	increment (most)	none (3)	include (3)		
last_sent	40	time message sent (most)	none (2)			
levels	32	level count (most)	game (8)	none (3)		
Lock	44	concurrency (most)	game (2)	none (1)	lake (1)	
magnitude	45	size of (most)	absolute value (5)	none (3)	game (3)	
mirror	45	copy/cache/backup (most)	none (6)	game (5)		
misses	34	count of (most)	game (6)	cache (4)	wife (1)	none (1)
near	39	close (11)	shortptr (8)	none (8)	game (2)	
numsegs	44	number of segments (most)	linux (4)	game (2)	none (1)	
origin	24	coordinates (most)	parent (1)			
outside	39	none (9)	flag (7)	game (4)	linux (1)	
pagenum	45	number (15)	document (14)	memory (3)	counter (3)	none (2)
picture	24	image (13)	pointer (3)	none (2)	Cobol (1)	
play	45	sound (14)	start something (11)	game (8)	none (2)	
position	35	location/coordinates (most)	in list (5)	game (5)		
purge	41	clean out (13)	delete (12)			
quick	44	flag (most)	none (10)	fast (9)	game (4)	optimization (3)
registered	45	flag (most)	registration (6)	signed on (4)	Linux (2)	game (1)
reliable	43	none (9)	trustworthy (5)	correct (4)	communication link (3)	game (2)
routine	22	function (9)	none (6)	ordinary normal (3)		
rover	32	none (14)	dog (6)	data structure (3)	car (3)	
self	44	this (18)	object (18)	game (6)		
single	35	none (8)	game (6)	singleton (5)	flag (3)	
stopped	44	process (11)	finished (6)	none (2)	game (2)	flag (1)
transformed	43	none (2)	game (2)	flag (1)	changed (1)	
translation	40	language (12)	cartesian (6)	none (4)		

The responses contained fewer different meaning per identifier than the Nickerson and Cartwright[?] study. However, this experiment did not explicitly request subjects to list all possible meanings of an identifier.

5 Discussion

This study set out to investigate the extent to which knowledge of the applicable application domain affected the meaning assigned to identifier names. A single experiment was performed, resulting in a single data point. More measurements, based on responses for other identifiers and application domains, are needed before it is possible to draw any general conclusions about the interaction between developer knowledge of the application domain and the meaning assigned to identifiers.

However, the 12.4% of game subject responses having a game context is significantly less than 100%. Some of the possible reasons for this include:

- subjects implicitly knew that many identifiers appearing in source code have no direct connection to the application domain. That is to say, many identifiers are used in the implementation of some algorithm and the choice of their names is primarily influenced by this algorithmic context. The meanings assigned to identifiers reflected this developer knowledge of typical identifier usage patterns,
- a failure by subjects to provide all of the information needed by this study. It is possible that the large number of identifiers appearing in the handout and the short amount of time available led to subjects deciding to provide brief, rather than detailed, responses. Subjects were not aware of the exact nature of the experiment or the kind of information it was hoped they would provide.

A *flag* meaning was given in a surprising number of responses. This may represent a default response, given when subjects could not think of anything else to write, or perhaps the identifier names used in this experiment often have this meaning in source code.

The responses generally involved concepts encountered in software engineering.

6 Conclusion

As the first of its kind the results of this experiment encountered a number of problems:

- feedback from subjects suggested that in the short space of time available they were not able to reliably estimate the quantity of code read/written. Given that few developers regularly measure the amount of source they have read/written it is not clear that anybody would be able to provide a reasonably accurate answer to this question,
- many of the written responses provided by subjects had a low information content (i.e., the question being asked was not answered). Providing subjects with more time and asking them to provide a detailed response, or interviewing subjects on a one-to-one basis would solve this problem,
- feedback from subjects suggested that without the context of the surrounding code it was difficult to provide what they considered to be a good interpretation of the likely meaning of an identifiers name,
- choosing identifiers based on their occurrence in various programs may prevent experimenter bias and provide a good justification for their use, but it severely restricts the semantic range of identifiers that can be used.

7 Further reading

In many ways identifiers are metaphors. For a fascinating introduction to metaphors in English see: "*Metaphors we live by*" by G. Lakoff and M. Johnson.

For an interesting, and readable, discussion of peoples performance in answering questions they do not know the answer to see: "*Simple heuristics that make us smart*" G. Gigerenzer and P. M. Todd.

The University of South Florida word association norms can be downloaded from: `w3.usf.edu/FreeAssociation`.

The responses given in the experiment (stripped of subject background information) can be downloaded from: `www.knosof.co.uk/cbook/accu2003.html`.

7.1 Acknowledgements

The author wishes to thank everybody who volunteered their time to take part in the experiment and the ACCU conference organizers for making conference slots available to run it.

References

At the end of each entry, the pages on which that entry is cited are listed in parentheses.

[1] anon. Justice standards clearinghouse. `it.ojp.gov/jsr/public/index.jsp`, 2004.

[2] J. M. Carroll. *What's in a Name? An essay on the psychology of reference*. W. H. Freeman, 1985.

[3] S. L. Ehrenreich and T. Porcu. Abbreviations for automated systems: Teaching operators the rules. In A. Badre and B. Shneiderman, editors, *Directions in Human/Computer Interaction*, chapter 6, pages 111–135. Ablex Publishing Corp., 1982.

[4] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. Statistical semantics: Analysis of the potential performance of key-word information systems. *The Bell System Technical Journal*, 62(6):1753–1805, 1983.

[5] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication: an analysis and a solution. *Communications of the ACM*, 30(11):964–971, 1987.

[6] K. W. Hirsh and J. J. Tree. Word association norms for two cohorts of British adults. *Journal of Neurolinguistics*, 14(1):1–44, 2001.

[7] W. Labov. The boundaries of words and their meaning. In C.-J. N. Bailey and R. W. Shuy, editors, *New ways of analyzing variation of English*, pages 340–373. Georgetown Press, 1973.

[8] D. L. Nelson, C. L. McEvoy, and T. A. Schreiber. The university of south Florida word association, rhyme and word fragment norms. Technical report, University of South Florida, Aug. 1999. `www.usf.edu/FreeAssociation`.

[9] C. A. Nickerson and D. S. Cartwright. An empirical thesaurus: Meaning norms for 90 common words, complete tables. Technical Report 85, University of Colorado at Boulder, Oct. 1979.