Developer beliefs about binary operator precedence (part 2 of 2)

Experiment performed at the 2006 ACCU Conference

First published in CVu vol. 19 no. x

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 2006 ACCU conference. The first part was published in a previous issue of C Vu.[12] This second part discusses the remember/recall assignment statement component of the experiment. See part 1 for a discussion of the experimental setup.

The format of the task performed in this part of the experiment is very similar to the memory for assignment statements portion of the experiment performed at the 2004 ACCU conference.[10] See the write-up of that experiment for some of the common details omitted here. That experiment attempted to measure the impact of identifier length, measured in syllables, on subjects' ability to remember assignment statement information over a short period of time. Experience with running the 2004 experiment showed that subjects sometimes used a strategy of remembering identifiers by storing information on their first letter rather than the complete identifier spelling. The identifiers used in the 2006 experiment were chosen to investigate the performance differences caused by identifiers sharing a common first letter and having a similar sounding spoken form.

The identifiers used in the 2006 experiment all contained letter sequences, having the form consonant-vowel-consonant, that could be spoken as a single syllable. For some problems all of the identifiers started with the same letter, while for other problems the initial letter of each identifier was different but the last two letters were the same.

Within commercial source code a variety of different kinds of character sequences are used for identifiers. Some are recognizable words or phrases, some abbreviated forms of words or phrases, while others have no obvious association with any known language (e.g., they may be acronyms that are unknown to the reader). Reading involves converting these character sequences to sounds and it is to be expected that subjects memories of an identifier will be sound based, rather than vision based.

2 Characteristics of human memory

The human *short term memory* subsystems are a gateway through which all conscious input data input must pass. They have a very limited capacity and because new information is constantly streaming through them, a particular piece of information rarely remains within them for very long. Information in short term memory is either quickly lost or stored in another, longer term, memory subsystem.

An experiment performed at the 2004 ACCU conference[10] investigated the impact on subject performance of identifiers that required more of less short term memory resources. Experience with this experiment suggested that subjects used a variety of strategies to help improve their performance. One strategy was to remember the first letter of an identifier, rather than a representation of the complete letter sequence. The identifiers used in this experiment were chosen to investigate the impact of shared letters on subject performance; they either share the same first letter on the last two letters (in this latter case the spoken forms rhyme).

The following are some of the factors that studies have been found to effect subject performance of memory for lists of information. These factors are also likely to have some impact on subject performance in this experiment.

- People pay particular attention to the initial part of a word[4, 15] (this enables them to start looking up a word in the mental lexicon while its remaining sounds are being heard).
- A decrease in word list recall performance for similar sounding words.[3, 5] It is believed that the similarity causes confusion between the various word sound sequences and a subsequent failure to correctly retrieve the original information.
- The extent to which the information to be remembered is already stored in longer term memory subsystems (i.e., known letter sequences such as words).
- The time delay between seeing the information and having to recall it (because the remembered information degrades over time),
- A capacity limit on the total amount of information that can be remembered and shortly afterwards recalled or recognized,

- For known words, their frequency of occurrence, with better performance in many tasks for high frequency words (i.e., those that have been encountered very frequently by a subject) compared to low frequency words.[14]
- Neighborhood effects.[2] Words that differ by a single letter are known as *orthographic neighbors*. Both the *density* of orthographic neighbors (how many there are — *mine* has 29 (*pine*, *line*, *mane*, etc.)) and their relative frequency (if a neighbor occurs more or less frequently in written texts) can affect performance.

Spotting the identifier that did not appear in the earlier list of assignment statements is a recognition problem, while remembering the value assigned in a recall problem. Studies have found that recognition and recall memory have different characteristics.[1]

3 Ecological validity

For the results of this experiment to have some applicability to actual developer performance it is important that subjects work through problems at a rate similar to that which they would process source code in a work environment. Subjects were told that they are not in a race and that they should work at the rate at which they would normally process code. However, developers are often competitive and experience from previous experiments has shown that some subjects ignore the work rate instruction and attempt to answer all of the problems in the time available. To deter such behavior during this experiment the problem pack contained significantly more problems than subjects were likely to be able to answer in the available time (two people did answer all problems).

The structure of the problem used in this experiment follows a pattern that is often encountered when trying to comprehend source code: see information (and try to remember some of it), perform some other task and then perform a task that requires making use of the previously seen information.

Taken as a whole the constant repetition of exactly the same kind of problem rarely occurs in program development activities. The constant repetition provides an opportunity for learning to occur, i.e., subjects have the opportunity to tune their performance for a particular kind of problem. The issue of learning and problem solving strategies used by subjects is discussed below.

4 Generating the assignment problems

The problems and associated page layout were automatically generated using a C program and various awk scripts to generate troff, which in turn generated postscript. The identifier and constant used in each assignment statement was randomly chosen from the appropriate set and the order of the assignment statements (for each problem) was also randomized. The source code of the C program and scripts is available from the experiments web page.[13]

Due to a fault in the generation script the first 10 problems for each subject all used sets of identifiers where the last two letters of each set of identifiers were the same. The intent was that the same randomisation algorithm be applied to the choice of identifiers to use for all problems.

4.1 Selecting identifiers and integer constants

A sufficient number of letter sequences were created so that subjects would rarely encounter the same sequence. In all 40 different words and 40 different nonwords were used (dues to an oversight *cub* appeared both in a set of words and a set of non-words), see the following word list. This meant that the same identifiers would start to repeat after every set of 20 problems.

The impact of different kind of letter sequences is the primary concern and we want to maximise the impact of differences due to this factor. This means minimising the impact of other kinds of information (mostly integer constants) on subject performance. A good approximation to short term memory requirements is the number of syllables contained in the spoken form of the information. Choosing single digit integer constants containing a single syllable minimises their impact on short term memory load.

For simplicity it was decided that identifiers would consist of a sequence of three letters following the pattern CVC. The letter sequences were grouped in sets of four such that, they were all either English words or pronounceable English non-words, and either:

- had different first letters and rhymed (i.e., in the last two letters were the same),
- shared the same first letter and did not rhyme (i.e., the last two letters did not share any common letters).

No checks were made for nonwords, in English, that might be words in other languages that might be known to subjects.

For the letter sequences used in this experiment there was no STM capacity advantage to remembering just the letter of a word. The spoken form of single letters are represented by a single syllable (except *w* which contains two) and each of the letter sequences used was pronounceable as a single syllable. However, there is a potential advantage to only remembering the first letter when the set of identifiers *sound alike*. Using this strategy would remove the possible confusion caused by similar sounding identifiers and eliminate a potentially significant source of performance loss.

The following lists the sets of four, three letter sequences used for identifiers appearing in the assignment part of each problem. The identifiers appearing in a single row were used to create one complete assignment problem. The letter sequence *cub* was mistakenly used in both a row of words and a row of non-words.

cat	mat	hat	pat
hen	pen	men	ben
din	pin	sin	kin
hop	pop	top	mop
cub	rub	tub	hub
dat	lat	wat	gat
gen	ren	sen	cen
nin	rin	zin	cin
dop	gop	vop	rop
fub	lub	wub	cub
dad	den	dip	dog
lap	led	lip	lot
pat	peg	pin	pod
sat	sir	sow	sum
wad	web	wit	won
fep	fis	fot	fum
kam	kig	kod	kus
ras	rit	roz	ruc
tep	tid	tor	tul
vek	vib	vom	vup

The nonwords have a variety of characteristics, including: *sen*/*cen* different spelling same spoken sound, *cin* sounds like *sin* a word, *fot* could be remembered as *foot* + CVC pattern, *roz* rozzer slang for policeman (at least in British English) or abbreviation for rosalyn. While these issues might be important at some level, the don't seem to have had a measurable impact on the results.

Assignment problems were created in groups of 20. Each group of 20 used one of the rows of identifiers. The identifiers used in each assignment problem were selected by randomly choosing an unused row. Three of the identifiers in the row were randomly selected to be used in the list of the three assignment statements to be remembered. The fourth identifier was used as the *not seen* identifier.

4.2 Selecting integer constants

The integer constants chosen were 4, 5, 6, 8, and 9 (the digit 7 was not used because its spoken form has two syllables). They all have approximately the same frequency of occurrence in source code (it is within

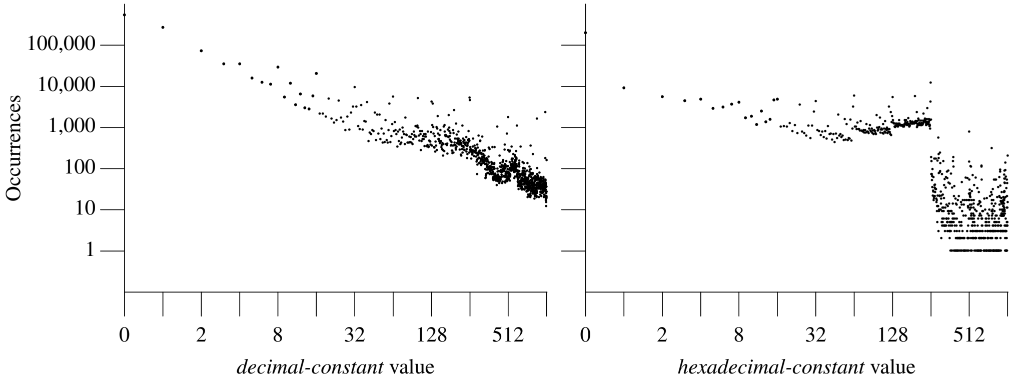

Figure .1: Occurrences, in the visible form of various applications written in C, of integer constants having specific values.[11]

the same order of magnitude, see Figure .1) and other contexts, and have a spoken form containing a single syllable.

5 Threats to validity

Experience shows that software developers are continually on the look out for ways to reduce the effort needed to solve the problems they are faced with. Because each of the problems seen by subjects has the same structure it is possible that some subjects will have detected what they believe to be a pattern in the problems which they attempt to use to improve their performance.

While the general format of problem used commonly occurs during program comprehension, the mode of working (i.e., paper and pencil) does not. Source code is invariably read within an editor and viewing is controlled via a keyboard or mouse. Referring back to previously seen information (e.g., assignment statements) requires pressing keys (or using a mouse). Having located the sought information additional hand movements (i.e., key pressing or mouse movements) are needed to return to the original source location. In this study subjects were only required to tick a box to indicate that they *would refer back* to locate the information. The cognitive effort needed to tick a box is likely to be less than would be needed to actually refer back. Studies have found[6] that subjects make cost/benefit decisions when deciding whether to use the existing contents of memory (which may be unreliable) or to invest effort in relocating information in the physical world. It is possible that in some cases subjects ticked the *would refer back* option when in a real life situation they would have used the contents of their memory rather than expending the effort to actually refer back.

While subjects were told that they are not in a race and that they should work at the rate at which they would normally process code, it is possible that some subjects followed this request and some did not. A consequence of this is that the distribution in the numbers of problems answered, and perhaps the accuracy of the results, may be different than would have occurred if all subjects had reacted in the same was to the instructions.

6 Results

It was hoped that at least 30 people (on the day 18) would volunteer to take part in the experiment and it was estimated that each subject would be able to answer 20 problem sets (on the day 23.8) in 20-30 minutes (on the day 20 minutes). Based on these estimates the experiment would produce 600 (on the day 429) individual answers.

A total of 429 sets of assignment statements were remembered/recalled giving a total of 1,716 answers to individual assignments. The average number of individual answers per subject was 95.3 (standard deviation 38.8), the average percentage of answers where the subject would refer back was 26.3% (standard deviation 26.7), and the average percentage of incorrect answers 8.9% (sd 9.5).

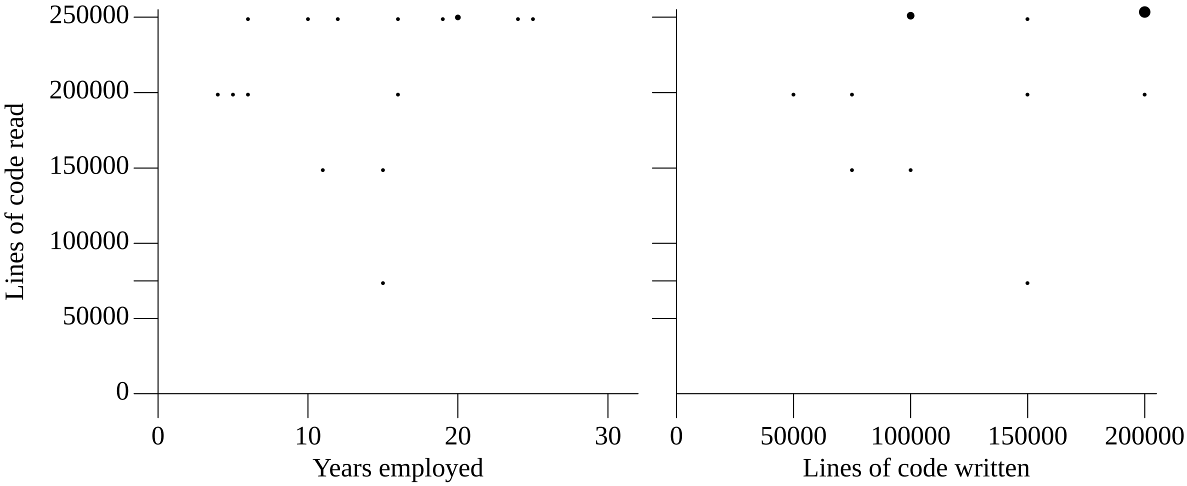

Figure .2: The plot on the left depicts number of line of code read against number of years of professionally experience. The plot on the right depicts the number of lines of code read against the number of lines of code written, for each subject. The size of the circle indicates the number of subjects specifying the given values. In those cases where subjects listed a range of values (i.e., 50,000-75,000) the median of that range was used.

The average amount of time taken to answer a complete problem was 50.4 seconds. No information is available on the amount of time invested in trying to remember information, answering the parenthesis sub-problem, and then thinking about the answer to the assignment sub-problem (i.e., the effort break down for individual components of the problem).

While STM recall performance drops very quickly after the information is no longer visible (studies have found below 10% correct within around 8 seconds in many situations[3]). Even the fastest subject took over 25 seconds per complete problem and so recency effects will be minimal.

The raw results for each subject are available on the experiments web page[13] (they are in the file `results.ans`; information on subject experience has been removed to help maintain subject anonymity).

The following subsections break down the discussion of results by individual subject and by kind of identifier used in the assignment statements.

6.1 Subject experience

Traditionally, developer experience is measured in number of years of employment performing some software related activity. However, the quantity of source code (measured in lines) read and written by a developer (developer interaction with source code overwhelmingly occurs in its written, rather than spoken, form) is likely to be a more accurate measure of source code experience than time spent in employment. Interaction with source code is rarely a social activity (a social situation occurs during code reviews) and the time spent on these activities is considered to be small enough to ignore. The problem with this measure is that it is very difficult to obtain reliable estimates of the amount of source read and written by developers. This issue was also addressed by studies performed at previous ACCU conferences.[9, 10]

It has to be accepted that reliable estimates of lines read/written are not likely to be available until developer behavior is closely monitored (e.g., eye movements and key presses) over an extended period of time.

Part 1 of this article contained a plot of precedence problems answered against years of experience. Given that a complete problem required subjects to answer both assignment and precedence problems this plot is actually a combined count of problems solved. No break-down is available on the time spent on the two different kinds of problem subjects were asked to answer.

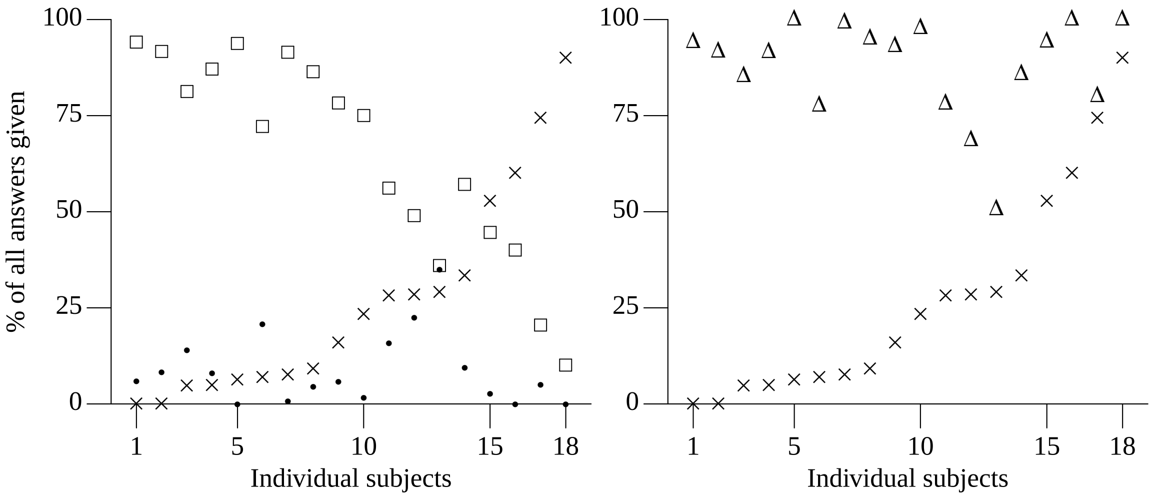

Figure .3: Number of problems answered by the 18 individual subject against the percentage of different kinds of answers. Left plot: cross for *would refer back*, bullet for incorrect answers and box for correct answers; right plot: cross for *would refer back*, triangle for percentage of correct answers for all cases where a numeric answer was given (i.e., *would refer back* answers were excluded from the percentage calculation). In both plots subjects, on the x-axis, are ordered by their *would refer back* percentage.

6.2 Subject strategies

Discussions with subjects who took part in the 2004 experiment uncovered that they had used a variety of strategies to remember information in the assignment problem. The analysis of the threats to validity in that experiment discussed the question of whether subjects traded off effort on the filler task in order to perform better on the assignment problem, or carried out some other conscious combination of effort allocation between the subproblems. To learn about strategies used during this experiment, after 'time' was called on problem answering, subjects were asked to list any strategies they had used (a sheet inside the back page of the handout had been formatted for this purpose).

The responses given to the strategies question generally contained a few sentences. The majority of responses dealt with the assignment part of the problem, with three subjects also giving information about the precedence problem (e.g., *I always use parentheses*, "... I tried to be consistent, but not very hard", "Didn't worry too much about task.").

The strategies listed consisted of a variety of the techniques people often use for remembering lists of names or numbers. For instance, number word associations, merging words into a larger word (e.g., penlenmen), reordering the sequence presented into a regular pattern (e.g., alphabetical), inventing short stories involving the words and numbers. The difficulty of integrating nonwords into these strategies was a common comment.

From the replies given it was not possible to work out if subjects give equal weight to answering both parts of the problem, or had a preference to answering one part of the problem.

No subject listed a strategy that was based on the visual appearance of the identifiers or numbers.

6.3 Individual subject performance

For each subject Figure .3 plots the percentage of each kind of answer they gave (in both graphs subjects are ordered by the percentage of *would refer back* answers they gave). The left plot is based on the percentages for all answers, while the triangles in the right plot are the percentage for those cases when a numeric answer was given (i.e., correct and incorrect answers only are used to calculate the percentage).

If subjects randomly guess answers to questions they cannot recall the answers to, then (given that only five possible numeric values were used and no value occurred more than once in the same problem):

- If subjects knew no answers and randomly guessed the three answers, then it would be expected that 0.7 of the three guessed questions (23%) of individual assignment questions would be answered correctly.

- If subjects knew one answer and randomly guessed the other two answers, then it would be expected that 0.5 of the two guessed answers (25%) for that problem would be answered correctly.
- If subjects knew two answers and randomly guess the last answer, then it would be expected to be correct 33% of the time.

Those subjects who gave few *would refer back* answers had a performance that was significantly better than that of random guessing. The analysis for those subjects who gave many *would refer back* answers and had a high percentage of correct answers is more complex. If these subjects randomly guessed the numeric answers they did give, the percentage correct would be very similar to that achieved when no *would refer back* answers were given. In this case the number of correct answers given by these subjects is significantly better than that of random guessing.

Possible reasons for this difference in performance, between subjects, include differences in subject's general approach to answering problems (e.g., in the case of *would refer back* the amount of risk they are willing to accept that the answer they are thinking of giving is incorrect) and differences in ability.

Looking at the right graph in Figure .3 we see that:

- subjects 1-8 were generally certain of the answer in the sense that they gave few *would refer back* answers (less than 23%) and that this certainty was mirrored in the significantly higher than random percentage of correct answers (average 91.8%),
- subjects 9-10 gave a higher percentage of *would refer back* answers than subjects 1-8, but also had a high percentage of correct answers (95.4%),
- subject 11 might be grouped with the first 10 subjects or subjects 12-13. This subject's percentage of correct answers is very close to that of subject 6 (78% vs. 77.6%), however the number of *would refer back* answers is more than four time greater (i.e., closer to that of subjects 12-13),
- subjects 12-13 gave *would refer back* answers to just over 25% of questions and the two lowest percentage of correct answers (68.5% and 50.7% respectively),
- subject 14, like subject 11, could belong to one of two subject groupings,
- subjects 15-18 gave *would refer back* answers to over 50% of questions. While these subjects also had a high percentage of correct answers (average 93.6%), this may be because they only gave answers in those cases where they were very certain (had they taken more risk they may have given more incorrect answers, or perhaps additional correct answers).

Self-knowledge, metacognition, is something that enables a person to evaluate the accuracy of the memories they have. Subjects who gave many incorrect answers (i.e., subjects 12 and 13) did not accurately evaluate the state of their own memories of previously seen information (i.e., they overestimated the accuracy of their memories). It is also possible that subjects who gave many *would refer back* answers also showed poor metacognitive performance (i.e., they underestimated the accuracy of their memories and would have mostly given correct answers had they risked a numeric answer). However, it is not possible to make this claim from the available data.

Were different subject's performance comparable through out the experiment? Perhaps a subject who answers a greater number of questions is more likely to give incorrect or *would refer back* answers. A least squares fit of the data (see Figure .4) suggests that subject's who answered more questions gave more *would refer back* responses and more incorrect answers. However, these results fail a statistical significance test at the 5% confidence level (i.e., it is not possible to claim that subjects who answered more questions gave more *would refer back* and incorrect answers).

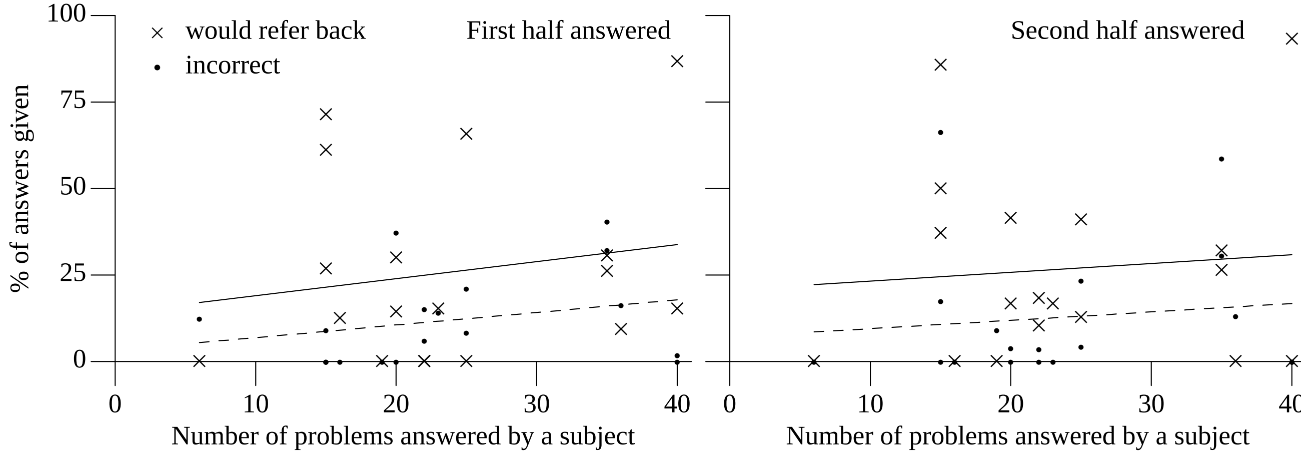

Figure .4: The percentage of *would refer back* answers (crosses, least squares straight line) and incorrect answers (bullet, least squares dashed line) plotted against the total number of problems answered by each subject. The left graph is based on the first half of the answers given by each subject, the right graph on the second half of the answers given. Each cross and bullet represents a single subject.

In the 2004 experiment there was no significant difference in performance between subjects who answered a few questions and those who answered many. However, the small number of unique identifiers used in the 2004 experiment and the ordering of the assignment statements both provided an opportunity for learning to occur as subjects answered more questions. In the 2006 assignment problems there does not appear to be any opportunity for subjects to improve their performance by learning as they answer more questions.

6.4 Performance changes over time

If subject performance was consistent for all problems answered, it would be expected that the percentage of correct answers for the first few problems answered would be the same as for the last few problems.

The data for Figure .4 was created by dividing the answers given by each subject into two equal sized parts, i.e., the first half of the answers given by each subject and the second half of the answers given. A difference in the slope of the least squares fit would indicate that subject performance changed over time. Unfortunately these results fail a statistical significance test at the 5% confidence level and it is not possible to draw any conclusions from differences in the slope of the least squares fit.

As previously stated the first 10 problems all used sets of identifiers that followed a single pattern. It is possible that subject performance for identifiers following this pattern was sufficiently large that it biased the results for the set of *first half* answered. It is also possible that there are significant effects involving both kinds of identifiers and early/late answers and that they cancelled each other out because a random ordering was not used.

6.5 Impact of different kinds of character sequences

This experiment was designed to look for differences in subject performance for different kinds of identifiers.

Each identifier appeared once per set of 80 assignment statements. Based on expected subject performance, it was anticipated that most identifiers would be seen once, with only a few identifiers being seem twice by the faster subjects. Thus there was no opportunity for learning of individual identifiers to take place.

Figure .5 gives a break down of performance for the different kinds of identifier. While it is tempting to try and read small differences in performance from these results, the variations are swamped by differences in individual subject performance (vertical bars denote the standard deviation for each average and they overlap significantly because the standard deviations are all relatively large).

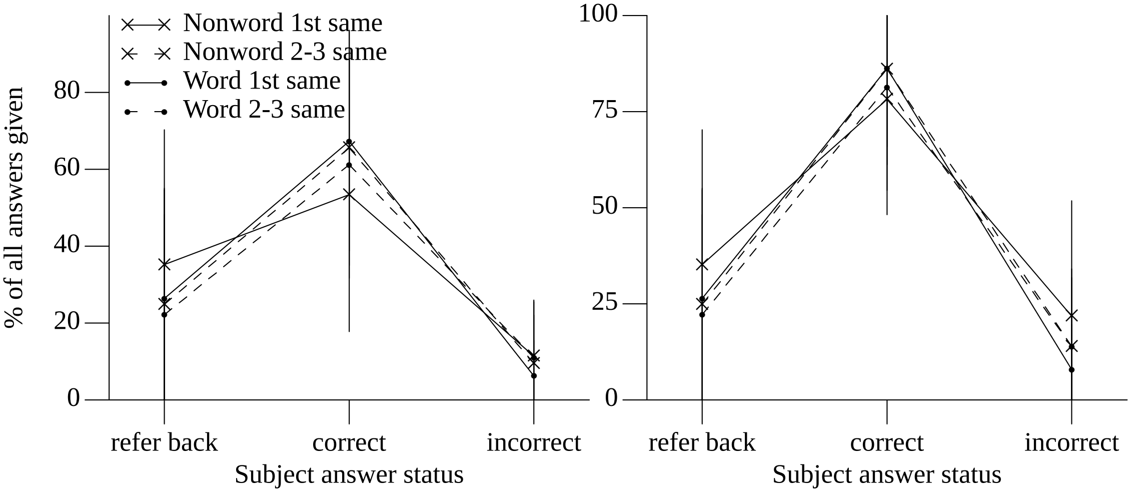

Figure .5: The percentage of *would refer back*, correct and incorrect answers for each kind of identifier, averaged over the individual respective percentage for all subjects. In the right plot, the percentage for correct and incorrect answers is based only on answers where a value was given (i.e., *would refer back* answers were excluded from the calculation). The vertical bars denote the standard deviation for each average (they overlap significantly because the standard deviations are all relatively large).

6.6 Comparison of 2006 results with 2004

How do the results of the 2004 and 2006 experiment compare? Both ran for 20 minutes and subjects completed an average of 22.7 problems in 2004 and 23.8 in 2006.

The *would refer back* percentages in 2004 were around 30%, correct answers around 60% and incorrect answers around 10%. These percentages are very close to the average percentages in 2006. Given that many of the 2004 identifiers contained three syllables (i.e., made greater calls on STM resources) the similarity between subject performances in the two experiments suggests that limited STM resources were not one of the primary factors affecting performance.

The filler problems used in the two experiments varied in the calls they made on cognitive abilities. The 2004 problem required holding information in STM and using it to solve an `if-statement` problem while the 2006 problem required making use of existing knowledge to solve a problem that only required a small amount of information to be held in STM.

7 Conclusion

Based on both years of employment and the claimed number of lines of code read/written the subjects taking part in the experiment had a significant amount of software development experience.

The number of years of software development experience is likely to have a high correlation with a subjects age. While cognitive performance has been found to decrease with age,[7, 8] age does not appear to have been a factor affecting the number of questions answered in this experiment (however, most subjects are likely to be younger than the age at which studies have found a significant age decrease in performance; i.e., 50s and over).

There was no significant difference in performance for the different kinds of identifiers used in this experiment. Any minor variations that might exist were swamped by differences in individual subject performance.

The most significant factors affecting assignment problem performance all seem to have their root in the mental characteristics of individual subjects. These characteristics are likely to include short term memory capacity limits, metacognitive (self-knowledge) ability, and degree of risk aversion.

Future experiments need to investigate whether subjects giving many *would refer back* answers have less ability of remember information or are not able to reliably evaluate the accuracy of the memories they have.

8 Further reading

For a readable introduction to human memory see “Essentials of Human Memory” by Alan D. Baddeley. A more advanced introduction is given in “Learning and Memory” by John R. Anderson. An excellent introduction to many of the cognitive issues that software developers encounter is given in “Thinking, Problem Solving, Cognition” by Richard E. Mayer.

9 Acknowledgments

The author wishes to thank everybody who volunteered their time to take part in the experiment and those involved in organising the ACCU conference for making a conference slot available in which to run it.

References

Citations added in version 1.0b start at 1449.

1. J. R. Anderson. *Learning and Memory*. John Wiley & Sons, Inc, second edition, 2000.
2. S. Andrews. The effect of orthographic similarity on lexical retrieval: Resolving neighborhood conflicts. *Psychonomic Bulletin & Review*, 4(4):439–461, 1997.
3. A. D. Baddeley. How does acoustic similarity influence short-term memory? *Quarterly Journal of Experimental Psychology*, 20:249–264, 1968.
4. H.-F. Chitiri and D. M. Willows. Word recognition in two languages and orthographies: English and Greek. *Memory & Cognition*, 22(3):313–325, 1994.
5. V. Coltheart. Effects of phonological similarity and concurrent irrelevant articulation on short-term-memory recall of repeated and novel word lists. *Memory & Cognition*, 21(4):539–545, 1993.
6. W.-T. Fu and W. D. Gray. Memory versus perceptual-motor tradeoffs in a blocks world task. In *Proceedings of the Twenty-second Annual Conference of the Cognitive Science Society*, pages 154–159, Hillsdale, NJ, 2000. Erlbaum.
7. A. S. Gilinsky and B. B. Judd. Working memory and bias in reasoning across the life span. *Psychology and Ageing*, 9(3):356–371, 1994.
8. D. Z. Hambrick and R. W. Engle. Effect of domain knowledge, working memory capacity, and age on cognitive performance: An investigation of the knowledge-is-power hypothesis. *Cognitive Psychology*, 44(4):339–387, 2002.
9. D. M. Jones. I_mean_something_to_somebody. *C Vu*, 15(6):17–19, Dec. 2003.
10. D. M. Jones. Experimental data and scripts for short sequence of assignment statements study. `http://www.knosof.co.uk/cbook/accu04.html`, 2004.
11. D. M. Jones. The new C Standard: An economic and cultural commentary. Knowledge Software, Ltd, 2005.
12. D. M. Jones. Developer beliefs about binary operator precedence. *C Vu*, 18(4):14–21, Aug. 2006.
13. D. M. Jones. Experimental data and scripts for developer beliefs about binary operator precedenc. `http://www.knosof.co.uk/cbook/accu06.html`, 2006.
14. M. Steyvers and K. J. Malmberg. The effect of normative contextual variability on recognition memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 29(5):760–766, 2003.
15. M. Taft and K. I. Foster. Lexical storage and retrieval of polymorphemic and polysyllabic words. *Journal of Verbal Learning and Verbal Behavior*, 15:607–620, 1976.