Measuring Programming Competence by Assessing Chunk Structures in a Code Transcription Task

Noorah Albehaijan[1,2] (N.albehaijan@sussex.ac.uk, naalbehaijan@iau.edu.sa)
Peter C-H. Cheng[1] (p.c.h.cheng@sussex.ac.uk)
[1]Department of Informatics, University of Sussex Brighton, BN1 9QJ, UK
[1,2]Department of Computer Science, Imam Abdulrahman Bin Faisal University Jubail, P.O. Box 12020, Saudi Arabia

Abstract

In a simple transcription task in which sections of Java program code are copied by freehand writing, it is demonstrated that chunk related temporal signals are sufficiently robust to permit the measurement of programming competence. An experiment with 24 participants revealed that the number of views of the stimulus per trial and the duration of writing per stimulus view are both strongly correlated with independent measures of Java competence.

Keywords: Chunking, program comprehension; competence measurement; transcription.

Introduction

Chunking (Miller, 1956; Cowan, 2001) underpins cognition in tasks that involve information of any complexity. Many phenomena are explained by the notion. For instance, at a long timescale, chunk acquisition explains many of the elevated abilities of experts over novices (e.g., Chase & Simon, 1973; Egan & Schwartz, 1979). At medium timescales, learning relies on the acquisition of chunks (Gobet et al., 2001). The organization of chunks changes during learning with the accretion of new chunks and the restructuring of networks of chunks. At short timescales, the structure of chunks in memory is one substantial factor in the control of routine sequential behaviour, such as the writing of memorised sentences (Cheng & van Genuchten, 2018) or the drawing of geometric diagrams (Obaidellah & Cheng, 2015).

All this suggests that it should be feasible to assess a learner's understanding or competence in a particular knowledge domain by evaluating behavioral measures that are dependent on the underlying structure of that learner's chunk network. And that such assessments can be done using simple production tasks, such as the written transcription of text or formulas, or the copying of diagrams.

Various studies have shown that certain measures of the distribution of the durations of inter-stroke pauses provide feasible measures of competence (Cheng, 2014, 2015; Cheng & Rojas-Anaya, 2007; van Genuchten et al., 2009; Zulkifli, 2013). An *inter-stroke pause* is the time that the pen is off the paper between written strokes, which provides measures at times scales in the range of 100 ms to 1 second . These studies typically used simple transcription tasks, in which the participants copied simple stimuli in each trial, such as a mathematical equations or one English sentence. Strong correlations with independent measures of domain comprehension were found. Further, the relative difficult of stimuli were clearly related to the magnitude of the pause measures. These findings were obtained across diverse domains (algebraic formulas and natural language), classes of users (children and adults) and interface media (pen on paper and on screen mouse driven symbol selection).

Pause measures in typewriting, keystroke logging, have been used extensively to study writing behaviour and performance (e.g., Spelman Miller & Sullivan, 2006), but this requires the aggregation of relatively large amounts of data in order to find effects. Also, our pilot experiments have shown that individual differences, such as variations in typing strategy and skill, tend to obscure the temporal chunk signals. So, inter-keypress pause measures do not appear to be reliable.

What other behaviors might provide strong and robust temporal chunk signals that can serve as a measure of comprehension? Can the scope of chunk-based measures of comprehension be extended to other domains beyond mathematics and natural language? The present experiment addresses these questions.

As chunking is important in the doing and learning of programming (e.g., Shneiderman, 1976; McKeithen, et al., 1981; Pennington, 1987), here we will focus on the assessment of learners' comprehension of programming code, specifically Java. Some studies have used response times to study programming comprehension in whole tasks, such as sets of multiple choice questions, lasting minutes (e.g., Adelson, 1981, 1984; Ye & Salvendy, 1996). Here, the focus is on the time required for component activities within a task, rather than overall task time, and the examination of process durations that may directly depend upon the chunks possessed by participants.

Again we will use a transcription task, as in the experiments cited above. In those experiments, typically, the stimulus was presented on a card or computer-screen placed near a writing tablet, so that the participants could switch their gaze between the stimulus and the tablet. In this experiment we will record when the participant switches between the stimuli and tablet using a participant-driven "hide-show" interaction method. The stimulus appears on the computer screen when the participant holds down a special button. To write the participant must release the button and the stimulus is masked. This extends the repertoire of techniques that may be used to assess chunk structures with a method that targets the processing of several chunks, at a 10 s timescale, which contrasts to the previous methods that analyse elements within a single chunk.

This method makes available various measures: (a) *view-numbers* – the total number of views of the stimulus in a trial;

(b) *writing-times* – the time spent writing between two successive views; (c) *view-times* – the duration of each look at the stimulus.

Various predictions can be derived for these measures. Experts perceive the stimuli using larger chunks than novices. Assume that working memory capacity for chunks does not vary substantially with expertise, which is plausible given that transcription is a relatively complex task (Cowan, 2001) rather than a simple decision making or capacity test (Miller, 1956). So, as the size of a stimulus is fixed, we predict:

H1) View-numbers*: the number of views of the stimulus in a trial will be less for more competent participants.*

As more competent participants' chunks contain more content, we predict:

H2) Writing-times*: the duration of written responses after each stimulus view will be longer for more competent participants.*

This assumes that writing speed is independent of expertise in the target domain, which is plausible for adult participants. Now, as the time to perceive a chunk is approximately constant (Chase & Simon, 1973), and if the number retained per view is independent of competence, then we predict:

H3) View-times*: the time spent on each separate view of the stimuli will not be directly related to competence.*

Frequently used components of Java are introduced earlier during instruction, so we predict:

H4) The performance on basic stimuli will be superior to advanced stimuli, with fewer view-numbers and longer writing-times, but no impact on view-time.

Note that H3 is framed negatively, so care is required to interpret data that might support it. In particular, the magnitude of other effects must be strong so that the likelihood of the absence of an overall view-time effect is not merely due to lack of statistical power. The underlying pattern of view-time data can also be examined for supporting evidence.

Clearly, the predictions depend on some strong assumptions, so unless the effects of chunking produce substantial temporal signals, no effective measures of competence will be obtained.

Method

The experiment was conducted at the University of Sussex with approval from the Science School's ethics committee.

Design

The experiment is a within participant design with each person transcribing basic and advanced sections of Java program code. The order of these trials was counter-balanced. The trials were preceded with two practice stimuli.

(Originally, the experiment was a counter-balanced 2X2 design with a fixed stimuli factor to provide pause distribution measures for comparison. Unfortunately, an obscure software-hardware interaction on the experimental computer was found during analysis. As the original counterbalancing does not appear to have affected the reported conditions, for clarity, the experiment is presented just as single factor.)

Participants

The participants were 24 adults from the School of Engineering and Informatics. Recruitment spanned first year undergraduate students through to members of faculty, to obtain good range of programming expertise. Age ranged from 19 to 59 years (*mean*=25, *SD*=8.51), and 15 were male and 9 females. They received £8 for participating.

```
#
public class Person{
    public String name;
    public int age;}

public void Balance(){
    System.out.println("#");
    Total += balance;}

int h=0;
while(h<hCount.length){
  System.out.println(h+hCount[h]);h++;}
```

Figure 1: Stimulus sample (basic).

Materials

The two practice stimuli consisted of series of simple statements, such as 'Computer Science', 'Programming Course', 'JAVA Programming Language'. Each of the four Java program code stimuli consisted of nine lines of code divided into three separate blocks. Each stimulus had an equal number of lines and the total number of characters differed by less than 5%. Figure 1 shows an example of one stimulus. Two *basic* and two *advanced* versions of the stimuli were created by consulting the course content of the student participants. The expressions in the basic stimuli were a core part of their JAVA instruction in their first year. The expressions in the advanced stimuli are more specialist items that would only have been seen by the better performing students.

The experiment was conducted using a standard graphics tablet (Wacom – Intuous3) connected to a PC running a logging program specially written in our lab. Participants wrote with an inking pen on a response sheet. The response sheet was printed with a grid of 17 lines; each consisting of 42 spaces for the writing of separate characters. The sheet was designed for non-cursive writing in order to provide rich inter-stroke pause data (see parenthetical note in the Design section). Participants adjust to this style of writing quickly and it does not appear to adversely affected other aspects of their performance (Cheng, 2014; Zulkifli, 2013).

Following the trials, the participants completed a questionnaire with four parts (on an internet survey platform). Part 1 included biographic questions relating to educational level. Part 2 assessed programming experience in general with five graduated rating items, such as 'I can develop programs using more than one object-oriented programming language". Part 3 assessed Java programming expertise level using eight graduated items, such as 'I am familiar with both objects and classes in Java'. Part 4 measured the participants' familiarity with the four specific Java stimuli that they were presented with during the trial. Participants were asked to judge what

their degree of familiarity would have been for each item *prior* to the experiment, on a 5 point Likert Scale.

Procedure

Participants were asked to hold the pen in their preferred hand and trained to: start writing at the beginning of each line, even for indented code; start writing as soon as the stimulus is revealed; copy the code as quickly and as accurately as they can; continue writing without correcting if they made a mistake; draw an upside down triangle symbol (inverted capital delta) in place of spaces; to start each trial with a hash (#); to hold down the special key to reveal with stimulus, with their preferred hand, which ensures that they write only when the stimulus key is released. The participants easily complied with these requirements and quickly became fluent in the practice trials. (Several of these conditions were needed for the pause measurements.) Similar trial requirements were successfully used in our previous experiments, so it is clear that they do not, on their own, undermine the reliability of the results.

For each trial, the response sheet was taped to the tablet. The participants finished the experiment within an hour.

Table 1: Correlation between competence measures. (N=24, Pearson correlation, 1 tail, critical value is 0.472 at $p<.01$)

	Education level	General programming	Java	Familiarity
Education level	–	0.366	0.183	0.181
General programming		–	0.759	0.734
Java			–	0.849
Familiarity				–

Results

Independent measure of competence

Questionnaire responses were coded to obtain independent competence measures against which to compare the chunk-based measures. Education level was scored on a scale from one to six (1=1st year undergraduate student, 6=faculty member). General programming and Java experience were scored by giving one point for each positive answer related to the measure, so had scales from zero to five and zero to eight, respectively. Ratings of the familiarity were scored from 0 (low) to 4 (high), so with the four stimuli, the overall scale runs from zero to twelve. Table 1 presented correlations between all combination of the measures, and is unsurprising. Education level is only weakly (and not significantly) correlated to the other measures. General programming experience has a strong positive relation to both Java experience and familiarity. The correlation between Java experience and familiarity are particularly strong. All this suggests that both Java experience or familiarity are specific to Java, rather than wider programming competence, and that either is suitable to serve as an independent measure. As the actual pattern of results is equivalent with either measure, just the analyses using familiarity are reported here.

Behavioural measures

The dependent behaviour measures were computed from the logs of each participant. The median writing-times and view-times were calculated for each trial. View-numbers is a count of interface switches to the stimuli (button presses). (We also computed a view related measure that discounted views of a stimulus without any accompanying writing before the next view, as some participants occasionally made such repeated views. The pattern of results using this measure is essentially the same as that with view-numbers.)

Figures 2, 3 and 4 show the total view-numbers, median writing-times and median view-times for participants rank ordered by their familiarity scores. Figures 5, 6 and 7 aggregate the data across low and high competent participants by showing the mean of the total view-numbers, the mean of the median writing-times and the mean of the median view-times. A binary split of participants' familiarity scores conveniently creates two equal size groups, with low scores exclusively below 6 or and high score exclusively above 8.

The first thing to note is that the total view-numbers, Figure 5, for the practice items is considerably lower than for the Java stimuli, but that the value is essentially equal at low and high competency (6.6 and 5.7, respectively). Similarly, the mean of the median writing-times, Figure 6, for the practice items is substantially longer than the Java stimuli, and although the value is greater for higher than lower competence (means of 14.2 and 12.1 s), it is not significantly so (by a *t* test; t=1.09, df=22, 1 tail, p=.24). These results reassuringly suggest that an effect of transcribing the Java stimuli exists beyond the act of merely transcribing any stimuli.

Consistent with prediction H1, Figure 5 shows that the high competence participants required fewer views than those with low competence, which is significant at both levels of stimuli (basic: 16.3 vs. 25.2, t=4.40, p=.0002; advanced, 20.0 vs. 28.5; t=4.05, p=.0005; both df=22, 1 tail).

Consistent with prediction H4, the basic stimuli demand fewer views than the advance stimuli across all participants (20.8. vs. 24.3; t=4.05, p=.0003; df=22, 1 tail). Further, for high competence participants the view-numbers is still significant despite the small group size (19.2 vs. 22.2; t=2.88, p=.016; df=10, 1 tail).

Consistent with prediction H2, Figure 3 and 6 show that the high competence participants had longer writing-times than those with low competence, which is significant at both levels of stimuli (basic: 10.7 vs. 6.5 s, t=3.86, p=.0008; advanced, 8.0 vs. 5.7; t=3.14, p=.005; both df=22, 1 tail).

Consistent with prediction H4, the advanced stimuli had shorter writing-times than the basic stimuli across all participants (6.9. vs. 8.6 s; t=3.29, p=.002; df=22, 1 tail). Further, for high competence participants the writing-time is still significant despite the small group size (8.0 vs. 10.7 s; t=3.7, p=.003; df=10, 1 tail), but not for the low competence participants (5.7 vs. 6.5 s, t=1.8, p=.1, df=10, 1 tail).

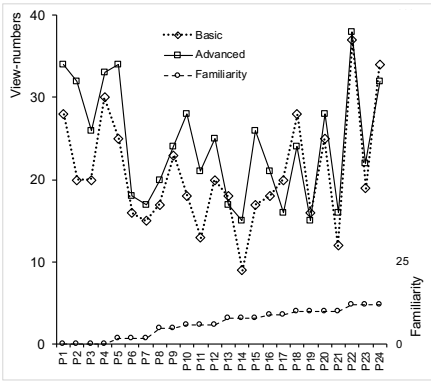

Figure 2. Total view-numbers for participants across basic and advance stimuli.

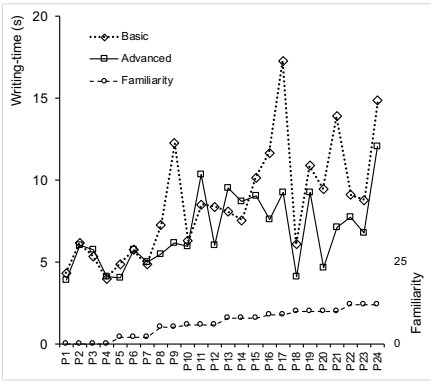

Figure 3. Median writing-times for participants across basic and advance stimuli.

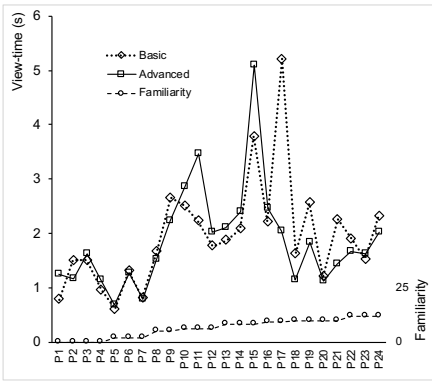

Figure 4. Median view-times for participants across basic and advance stimuli.

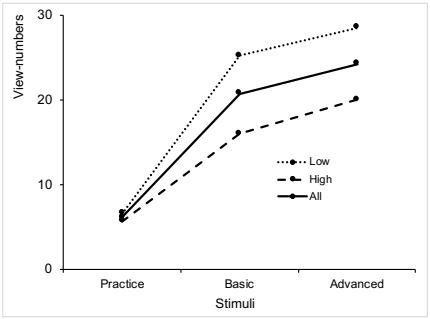

Figure 5: Mean view-numbers across stimuli type and level of competence.

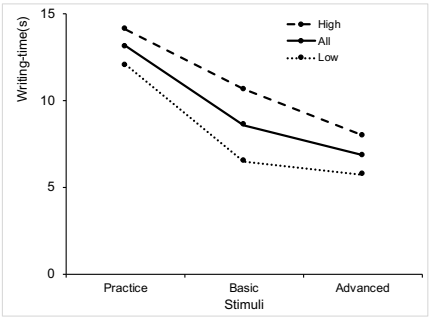

Figure 6. Mean of median writing-times across stimuli type and level of competence.

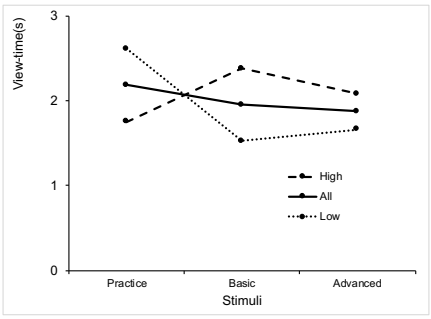

Figure 7. Mean of median view-times across stimuli type and level of competence.

Turning to H3, which concerns the absence of an overall effect of view-times, Figure 4 does not show a clear overall upward or downward trend in view-times, for both levels of stimuli difficulty. If anything, the overall pattern is an inverted 'u', in contrast to the trends in Figure 2 and 3. Figure 7 reveals that high competence participants have longer view-times than those with low competence, but this is not significant for the advanced stimuli (2.1 vs. 1.7 s; t=1.50, p=.15, df=22, 1 tail), but is marginally significant for the basic stimuli (2.4 vs. 1.5 s; t=2.62, p=.02, df=22, 1 tail). Further, comparing the view-times on the practice stimuli with the Java stimuli view-times we see they are similar, whereas for view-numbers and for writing-times the practice values are quite different to the Java stimuli values, as noted above.

Consistent with prediction H4, Figure 4 shows that nearly equal numbers of participants had longer view-times for basic stimuli or for advanced stimuli. In terms of the means across all participants, Figure 7, no significant differences occur for the basic stimuli (1.5 vs. 1.7, t=1.03, p=.3, df=22, 1 tail) nor the advanced stimuli (2.4 vs. 2.3; t=1.21, p=.25; df=22, 1 tail).

In summary, with respect to total view-numbers, means writing-times and view-times, all the predictions are supported.

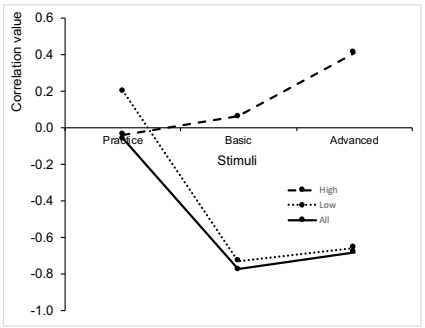

Figure 8: Correlation of view-numbers with familiarity across stimuli and competence.

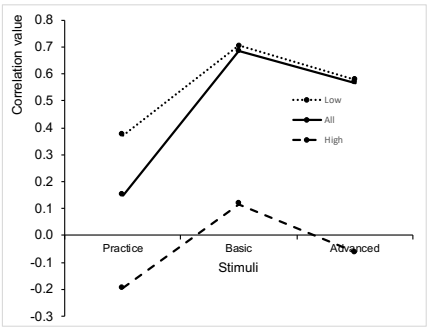

Figure 9: Correlation of writing-times with familiarity across stimuli and competence.

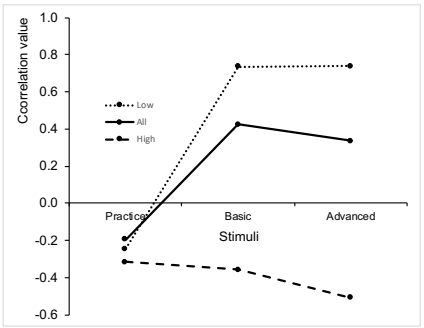

Figure 10: Correlation of view-times with familiarity across stimuli and competence.

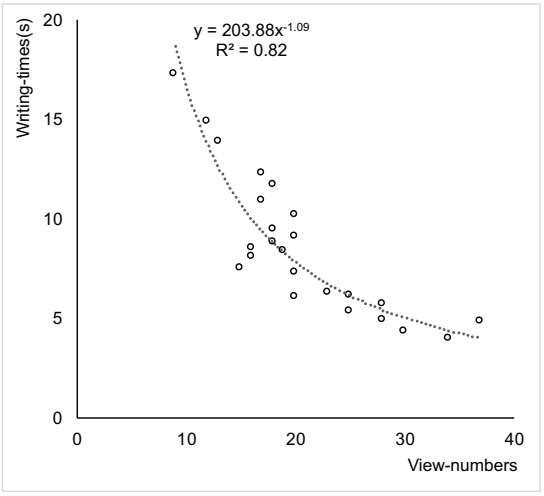

Figure 11: Relation of writing-times to view-numbers (basic stimulus)

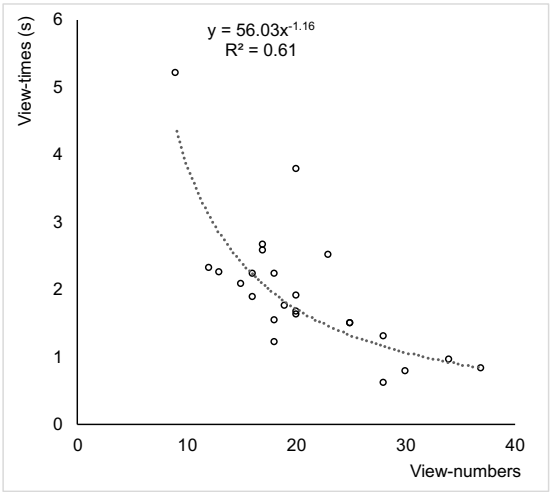

Figure 12: Relation of view-times to view-numbers (basic stimulus)

Correlation values for various measures

The correlations between view-numbers, writing-times and view-times versus familiarity were computed in order to further examine our four predictions. Figures 8, 9 and 10 show the Pearson correlations of familiarity score with, respectively, view-numbers, writing-times and view-times. The scale ranges are not the same. For correlations over all participants (solid line in Figs. 8-10) the critical value is 0.344 for significant correlations at $p<.05$, and 0.472 at $p<.01$ (1 tail, df=22). For correlations with just high competence or low competence participants (dashed or dotted lines) the critical value is 0.497 at $p<.05$ and 0.658 at $p<.01$ (1 tail, df=10).

As expected, none of the correlations for the practice items are significant. With view-numbers, Figure 8, across all participants the negative correlations are strong and significant: numbers of views decrease with competence (H1). The result is similar when just the low competence group is considered, but correlation for the high competence participants is positive but not significant. For writing-times the pattern of results is similar but the direction of the correlations are reversed, Figure 9: writing-time increases with competence (H2). For the whole group and the low competence subgroup the correlation for advanced stimuli is less than for the basic stimuli.

The view-times correlation, Figure 10, for the whole group and the high competent sub-group are not significant, but the correlations of the low competence participants are strong for both Java stimuli.

In summary, correlations for the view-numbers, writing-times and view-times are consistent with our four predictions, overall, but with some divergence in detail. In particular, view-numbers and writing-times did not differentiate high competence participants. Also, view-times did unexpectedly differentiate low competence participants, who needed more view time with increasing competence.

View-numbers vs. writing-times and view-times

The relation between our three main behavioural measures are examined because a systematic relation between them could provide further support for the hypotheses and more precise chunk-based explanations of the results. View number and writing-time are both predicted to be dependent upon chunking processes, so there should be some consistent and systematic relation between them. View-time is not expected to be chunk dependent, so no regular relation between it and view-numbers (or writing-duration) is anticipated. Scatter plots of these variables were drawn for all the participants in all the conditions of the experiment. Figure 11 plots writing-times versus view-numbers for the basic stimuli and Figure 12 is similar but for view-times. The pattern of data in Figure

Table 2. Parameter of best-fit power relation for writing-times and view-times to view-numbers

	Writing-times vs. view-numbers			View-times vs. view-numbers		
	Practice	Basic	Advanced	Practice	Basic	Advanced
Index, *i*	-0.95	-1.09	-0.97	-1.05	-1.16	-1.24
Constant, *C*	57.9	203.9	136.5	8.9	56	83.7
R-squared	0.459	0.818	0.747	0.603	0.615	0.623

11 has a particularly distinctive form, which is also apparent in the graph for the advanced stimuli. Thus, a power law curve for an inverse proportional relation was fitted to the data: the parameters of the best fit equations are given in Table 2, along with the R^2 values. The quality of fit for other equation forms (e.g., linear) were worse than a power law.

The power law for writing-time versus view-numbers is noteworthy, across both stimuli: the index is close to minus one and the R^2 values are large. This implies that the data is governed by a direct inverse proportional law. The relation between the view-times and view-numbers is less clear, with an absolute value of the index further from unity and lower R^2 values.

In other experiments, as yet unpublished, we have found similar patterns in view-numbers and writing-times data that closely fit an inverse proportional power law, so we are confident that the pattern is not accidental.

In summary, there appears to be a simple relation between the view-numbers and writing-times: a participant who takes twice the view-numbers of another will use half the time each time they write. But this simple relation does not hold for view-times.

Discussion

Previous studies have shown that measures of the distribution of inter-stroke pauses, captured in a simple transcription task, appear to reflect the different chunk structures of learners and hence may be used to assess the competence of the learners (Cheng, 2014; 2015, van Genuchten & Cheng, 2010; Zulkifli, 2013). This experiment extends those findings, in three ways.

First, allowing the user to reveal the stimulus at will, and hiding it during writing, allows two alternative temporal chunk measures to be captured: view-numbers — the total number of views of the stimulus in a trial; writing–times – the median duration of writing time between views. Predictions H1, H2, and H4 associated with these measures are well supported by converging evidence. The measures strongly correlated with our independent measures of competence. Further, no support for view-times as a suitable measure of competence was obtained, as predicted in H3, despite the relative strength of the effects for the other two measures.

Second, the experiment has shown that measures based on temporal chunk signals are applicable beyond mathematics (algebraic formula) and natural language, in a domain that happens to share some characteristics of both those domains.

Third, in contrast to the single line stimuli used in the previous experiments mentioned above, the present stimuli were larger (nine lines). The greater amount of data per trial means that single trials can yield strong usable correlations with competence, without the theoretical problems of deciding how to aggregate data from multiple trials or the practical problems associated with switching between multiple trials.

The overall correlations of view-numbers and writing-times with competence are strong, and this also holds for the low competence group. However, we must consider two qualifications. First, it is clear from Figures 2 and 3, that there is considerable variability between participants, such that some of the best low competence participant have better scores than many of the high competence participants, and vice versa. Clearly the development of a real educational test of programming competence must address the accuracy and sensitivity of the measures, perhaps by combining measures. Second, the curves in Figure 2 and 3 suggest that the view-numbers and writing-times may have plateaued for the high competent participants; in other words the difficulty of the advance stimuli may be insufficient to differentiate those within that group. This seems plausible, in hindsight, as the range of difficulty captured in the stimuli design was based on the undergraduate Java programming curriculum, but a proportion of the participants were drawn from more senior groups. This plateauing was also seen in previous studies (Cheng, 2014, 2015). One implication of this is the importance of designing stimuli with a sufficient range for the target test group.

The clear inverse proportional relation between writing-times versus view-numbers (Figure 11, Table 2) supports the chunk based explanations underpinning the predictions H1 and H2, and the poor fit of such a power law for view-times versus view-numbers is consistent with prediction H3. In particular, this implies that assumptions made for the predictions concerning the variability in participants working memory capacity and speed of writing are relatively small effects in comparison to chunk size variability with competence. In other words, the primary process in the transcription tasks appears to be the selection of chunks from the stimulus, with more competent participants retaining more characters – because they possess larger chunks – and this determines that time required for writing is in a direct proportion to the number of characters. Nevertheless, Figure 2 and 3 show much individual variability, so a useful line for future work is to investigate the possibility of separately measuring working memory capacity and writing speeds of participants in order to consider whether there is a need to devised methods to normalize for them.

Two observed effects might be spurious results, but they are sufficiently striking to deserve fuller investigation in further work. The first is the positive correlations of view-times with competence, specifically for low competence participants, is counter to prediction H3, Figure 10. The second is the increase in view-times with decreasing view-numbers, Figure 12: theoretically, there ought to be little relation between the two. One approach to study these effects is to probe the contents of participants' individual sets of chunks, which we are currently doing by extracting the locations of onset of views from the written logs in order to identify the precise content of participants' chunks. Our current hypothesis is that view-time variations may be due to differences in stimuli encoding strategies that fluctuate with content type.

This paper contributes a method for evaluating competence in a programming using a transcription task and measures with timescale of 10 s. This extends the range of techniques beyond the pause distribution measures of previous work (e.g., Cheng, 2014, 2015; Cheng & Rojas-Anaya, 2007).

Uses of the technique in education are readily imagined that exploit the relative simplicity, short trial times and the potential for fully automated scoring. Simply, such transcription tasks might be administered as a component of summative end-of-course evaluations or as standalone screening tests at the outset of a course. More interestingly with appropriately designed test items, the approach might be used as a form of formative assessment to provide tutors with information about individuals' growing understanding of targeted programming concepts. We are planning work on the development of the approach as a tool for use in computer-based tutoring systems.

Acknowledgments

Noorah's PhD study is supported by Imam Abdulrahman Bin Faisal University.

References

Adelson, B. (1981). Problem solving and the development of abstract categories in programming languages. *Memory & Cognition*, *9*(4), 422–433.

Adelson, B. (1984). When novices surpass experts: The difficulty of a task may increase with expertise. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *10*(3), 483–495.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology*, *4*(1), 55–81.

Cheng, P. C-H. (2014). Copying equations to assess mathematical competence: An evaluation of pause measures using graphical protocol analysis. In *Proc. of the 36th Annual Conf. of the Cognitive Science Society*, 319–324.

Cheng, P. C-H. (2015). Analyzing chunk pauses to measure mathematical competence : Copying equations using 'centre-click' interaction . In *Proc. of the 37th Annual Conference of the Cognitive Science Society. Cognitive Science Society., Austin, TX*, 345–350.

Cheng, P. C-H., & Rojas-Anaya, H. (2007). Measuring Mathematical Formula Writing Competence: An Application of Graphical Protocol Analysis. In *Proc. of the Twenty Ninth Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society, 869-874

Cheng, P. C-H., & van Genuchten, E. (2018). Combinations of simple mechanisms explain diverse strategies in the free-hand writing of memorised sentences. *Cognitive Science, 42*, 1070–1109.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Science, 24*(1), 87-114.

Egan, D. E., & Schwartz, B. J. (1979). Chunking in recall of symbolic drawings. *Memory & Cognition*, *7*(2), 149–58.

Gobet, F., Lane, P. C. R., Croker, S., Cheng, P. C.-H., Jones, G., Oliver, I., & Pine, J. M. (2001). Chunking mechanisms in human learning. *Trends in Cognitive Science*, 5(6), 1236-1243.

McKeithen, K. B., Reitman, J. S., Rueter, H. H., & Hirtle, S. C. (1981). Knowledge organization and skill differences in computer programmers. *Cognitive Psychology*, *13*(3), 307–325.

Miller, G. A. (1956). The magical number seven plus or minus two: Some limits on our capacity for information processing. *Psychological Review, 63*, 81-97.

Obaidellah, U. H., & Cheng, P. C.-H. (2015). The role of chunking in drawing Rey complex figure. Perception and Motor Skills, 120(2), 535-555.

Pennington, N. (1987). Stimulus structures and mental representations in expert comprehension of computer programs. *Cognitive Psychology*, *19*(3), 295–341.

Shneiderman, B. (1976). Exploratory experiments in programmer behavior. *International Journal of Computer & Information Sciences*, *5*(2), 123–143.

Spelman Miller, K. & Sullivan, K. P. H. (2006). Keystroke logging: an Introduction. In G. Rijlaarsdam (Series Ed.) and K.P.H. Sullivan, & E. Lindgren. (Vol. Eds.), *Studies in Writing, Vol.18, Computer Keystroke Logging: Methods and Applications*. Oxford; Elsevier.

van Genuchten, E., & Cheng, P. C.-H. (2010). Temporal Chunk Signal Reflecting Five Hierarchical Levels in Writing Sentences. In *Proc. of the 32nd Annual Conference of the Cognitive Science Society. Austin, TX: Cognitive Science Society,* 1922-1927.

van Genuchten, E., Cheng, P. C.-H., Leseman, P. P. M., & Messer, M. H. (2009). Missing working memory deficit in dyslexia: Children writing from memory. In *Proc. of the 31st Annual Conference of the Cognitive Science Society (Pp. 1674-1679). Austin, TX: Cognitive Science Society*.

Ye, N., & Salvendy, G. (1996). Expert-novice knowledge of computer programming at different levels of abstraction. *Ergonomics*, *39*(3), 461–481.

Zulkifli, M. (2013). *Applying Pause Analysis to Explore Cognitive Processes in the Copying of Sentences by Second Language Users*. University of Sussex (Unpublished PhD Thesis).