The Effects of Enhanced Compiler Error Messages on a Syntax Error Debugging Test

BRETT A BECKER* UNIVERSITY COLLEGE DUBLIN

KYLE GOSLIN, GRAHAM GLANVILLE CCT COLLEGE DUBLIN

*BRETT DOT BECKER AT UCD DOT IE

The Problem With Compiler Error Messages

- One of the many challenges novice programmers face from the time they write their first program is inadequate compiler error messages.
- These messages are:
  - Immediate
  - Often the only feedback students get
  - Essential for correcting code (of course)
  - Come from a machine that students think is infallible
- But they are very often…

The (50+ year old) Problem With Compiler Error Messages

inadequate and not understandable (Moulton and Muller, 1967), **useless** (Wexelblat, 1976), **not optimal** (Litecky and Davis, 1976), **inadequate (again, 17 years later)** (Brown, 1983), **frustrating** (Flowers *et al.*, 2004), **cryptic and confusing** (Jadud, 2006), **notoriously obscure and terse** (Ben-Ari, 2007), **undecipherable** (Traver, 2010), **still very obviously less helpful than they could be** (McCall and Kölling, 2014), **inscrutable** (Ko, 2017), and **a source of frustration and a barrier to progress** (Becker *et al.*, 2016).

See Becker *et al*. (2016) for more, and for references.

***Becker *et al.* (2016)** Brett A. Becker, Graham Glanville, Ricardo Iwashima, Claire McDonnell, Kyle Goslin & Catherine Mooney. Effective compiler error message enhancement for novice programming students, Computer Science Education, DOI: 10.1080/08993408.2016.1225464

Recent Results on Enhancing Compiler Error Messages

- Increased activity since ~2010
- Some seemingly conflicting results
- Different studies have attempted to measure the effects of enhanced compiler error messages on different metrics:
  - Error frequency, overall number of errors, repeated errors, errors per student
  - Attempts required to resolve errors
  - Non-compiling submissions
  - Student progress, test scores, academic performance
  - Time to resolve errors

Approach

- Most prior studies measure how many errors students **produce**, and then possibly rectify while writing programs (or some derived metric based on this)
- We measured how many **pre-existing syntax errors are rectified** by students while debugging programs
- Specifically, we measured the effect of enhanced compiler error messages in an empirical control/intervention experiment where students were given the task of removing syntax errors from non-compiling source code they did not write.

Methodology

- 86 CS1 (BSc) students studying Java in separate groups
- Utilized custom Java editor that can enhance error messages
  - Half of students received standard javac error messages
  - Other half received enhanced error messages
  - Both groups took the same debugging test under the same conditions

The Test

- ~100 (fairly sparse) lines of code, available at www.brett.becker.com/sigcse18/
- 24 occurrences of 11 different syntax errors (some appeared more than once)
  - Distribution intended to mimic actual distribution (some common and frequent errors, some more rare and infrequent errors)
- Test divided into 5 logical tasks with known, pre-specified output (once errors are rectified)
  - This attempted to mitigate for illegitimate fixes (e.g. deleting line of code with error)

The Test

Compiler error message type	Quantity	Rank	Frequency
cannot find symbol	4	1	19%
';' expected	4	3	12%
not a statement	4	4	9%
illegal start of expression	3	5	7%
bad operand types for binary operator *op*	1	8	5%
incompatible types	2	9	4%
<identifier> expected	1	10	4%
variable *v_name* is already defined in method *m_name*	2	13	3%
']' expected	1	16	<2%
missing return statement	1	*17*	<2%
package *p_name* does not exist	1	25	<1 %

- Compiler errors included in test
  - **Quantity** is the number of errors in each test generating the corresponding compiler error message type
  - **Rank** is the overall frequency rank from control group* (1 is most frequent)
  - **Frequency** is the frequency, from control group*

*__Becker (2016)__ Brett A. Becker, 2016, An Effective Approach to Enhancing Compiler Error Messages, In Proceedings of the 47th ACM Technical Symposium on Computer Science Education (SIGCSE '16), ACM, 126–131. DOI: http://dx:doi:org/10:1145/2839509:2844584

The Test

- In analyzing the submissions, we tabulated the following:

1. whether the submission compiled
2. if the program compiled, whether the output of each of the five tasks was correct
3. whether each of the 24 errors were rectified in a manner consistent with the specified output
4. a student score based on number of errors rectified

Research Questions

- RQ1: Do enhanced compiler error messages have an effect on the **number of successfully compiling submissions**?
- RQ2: Do enhanced compiler error messages have an effect on the **number of tasks completed correctly**?
- RQ3: Do enhanced compiler error messages have an effect on the **number of rectified syntax errors**?
- RQ4: Do enhanced compiler error messages have an effect on **student scores**?

Results

- RQ1: Do enhanced compiler error messages have an effect on the number of **successfully compiling submissions**?

- The intervention group submitted 20 compiling submissions and the control group submitted 17. A chi-squared test indicated that the difference between groups is not significant
  $\chi^2 = 0.43, p = 0.513\ (\alpha = 0.05)$

Results

- RQ2: Do enhanced compiler error messages have an effect on the **number of tasks completed correctly**?

Task	Control	Intervention	χ^2	p
A	10	10	-	-
B	12	13	0.06	0.813
C	13	12	0.06	0.813
D	13	13	-	-
E	10	12	0.24	0.621

- No differences are statistically significant

Results

- RQ3: Do enhanced compiler error messages have an effect on the **number of rectified syntax errors**?

- The control group rectified 665 (64.4%) of their errors while the intervention group rectified 790 (76.6%). A chi-squared test indicated that this is statistically significant $\chi^2 = 36.40, p < 0.001$ $(\alpha = 0.05)$

Results

- RQ3: We next sought to determine which compiler error messages contributed to this result.

Compiler Error Message Type	Total Errors	Rectified (Control)	Rectified (Intervention)	χ^2	p	**Sig(B-H)?**	Sig*?
cannot find symbol	172	121	145	9.55	0.003	**Y**	Y
';' expected	172	111	131	5.57	0.018	**Y**	N
not a statement	172	116	135	5.32	0.021	**Y**	Y
illegal start of expression	129	87	95	1.19	0.275	**N**	Y
bad operand types for binary operator *op*	43	23	28	1.20	0.272	**N**	N
incompatible types	86	57	70	5.09	0.024	**Y**	Y
<identifier> expected	43	23	30	2.41	0.121	**N**	Y
variable *v_name* is already defined in method *m_name*	86	49	57	1.57	0.210	**N**	Y
']' expected	43	35	37	0.34	0.559	**N**	N/A
missing return statement	43	13	23	4.78	0.029	**Y**	N/A
package *p_name* does not exist	43	30	39	5.94	0.015	**Y**	N/A

Sig (B-H)? indicates statistical significance (or lack of) with Banjamini-Hochberg correction for multiple tests

*__Becker (2016)__ Brett A. Becker, 2016, An Effective Approach to Enhancing Compiler Error Messages, In Proceedings of the 47th ACM Technical Symposium on Computer Science Education (SIGCSE '16), ACM, 126–131. DOI: http://dx:doi:org/10:1145/2839509:2844584

Results

- RQ4: Do enhanced compiler error messages have an effect on **student scores**?

- We calculated a score per student based on the number of errors corrected (perfect score = 24). The average/median scores were 15.16/16.5 (control) and 17.95/21 (intervention).

  - A Shapiro-Wilk test revealed neither distribution was normal.

  - A Mann-Whitney *U* test (two-tail, independent samples) showed no statistically significant difference between groups $U = 805.5, p = 0.172$ $(\alpha = 0.05)$

Discussion

- RQ1: Do enhanced compiler error messages have an effect on the **number of successfully compiling submissions**? **No**
- RQ2: Do enhanced compiler error messages have an effect on the **number of tasks completed correctly**? **No**
- RQ3: Do enhanced compiler error messages have an effect on the **number of rectified syntax errors**? **Yes**
  - 6 error messages showed a statistically significant reduction in number of rectified errors, and 3 of these also showed an effect in Becker (2016)
- RQ4: Do enhanced compiler error messages have an effect on **student scores**? **No**

Threats to validity

- Possible that a given error could be successfully resolved in a number of ways
- It is probable that a different test, but with the same number of the same syntax error message types, would generate different results
- Evidence that fewer control students attempted the final third of test*
  - Prather et al. (2017) observed similar behavior
  - The enhanced error messages may explain this
  - This also may be a reason that an effect was observed for some error messages

*__Prather et al. (2017)__ James Prather, Raymond Pettit, Kayla Holcomb McMurry, Alani Peters, John Homer, Nevan Simone, and Maxine Cohen. 2017. On novices' interaction with compiler error messages: A human factors approach. In Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER '17). ACM, 74–82. DOI:h.p://dx:doi:org/10:1145/3105726:3106169

Conclusions

- This study seemed to solidify one possible view on the current state of compiler error enhancement:
  - Some studies agree, others disagree
  - Some metrics that are ‘close to the data’ such as error frequency seem to be affected by enhanced compiler error messages
  - Other derived metrics that are ‘further from the data’ seem to not be affected by enhanced compiler error messages

Moving Forward

- Back to the drawing board?
  - Study the readability of compiler error messages?
  - Do we need to apply more rigor in designing enhanced compiler error messages?

Thank you!

Questions?