

Recognizing Opportunities for Mixed-Initiative Interactions in Novice Programming

Liam Doherty, Patrick Lougheed, David Brokenshire, Mayo Jordanov, Shilpi Rao,
Jurika Shakya, and Vive Kumar
Simon Fraser University, Surrey Campus

introduction

- learning programming is difficult
- novice programmers need help
- ▶ how do we help the users?

helping users

- novice programmers usually understand syntax at the lexical level
- we want them to start understanding the syntax at the pragmatic and semantic level
- usually, we have only the results of their programming exercises
- to help them better we need to see their process and interactions
- ▶ using formative data, we can help the novices while they are learning
 - need to present information to them at the right time

getting formative data

- compile time segments (CTSs)
 - differentials between consecutive versions of the AST
 - generated at each compile cycle
 - compiler output
- ▶ allow us to find errors and identify areas where user needs help

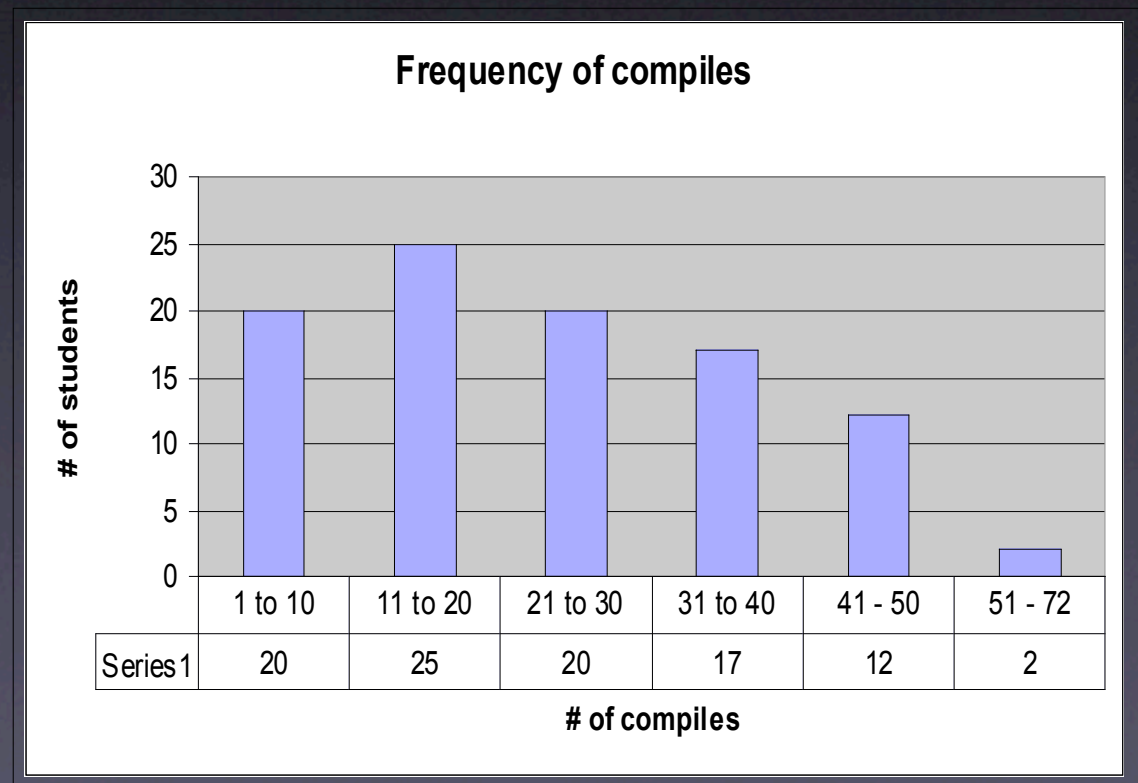
types of data

- compile time analysis of CTSs
- differentials in constructs
- compiler output
- construct differentials and compiler output

compile time analysis of CTSs

- number of compiles
- LOC
- number of errors
- degree of completion

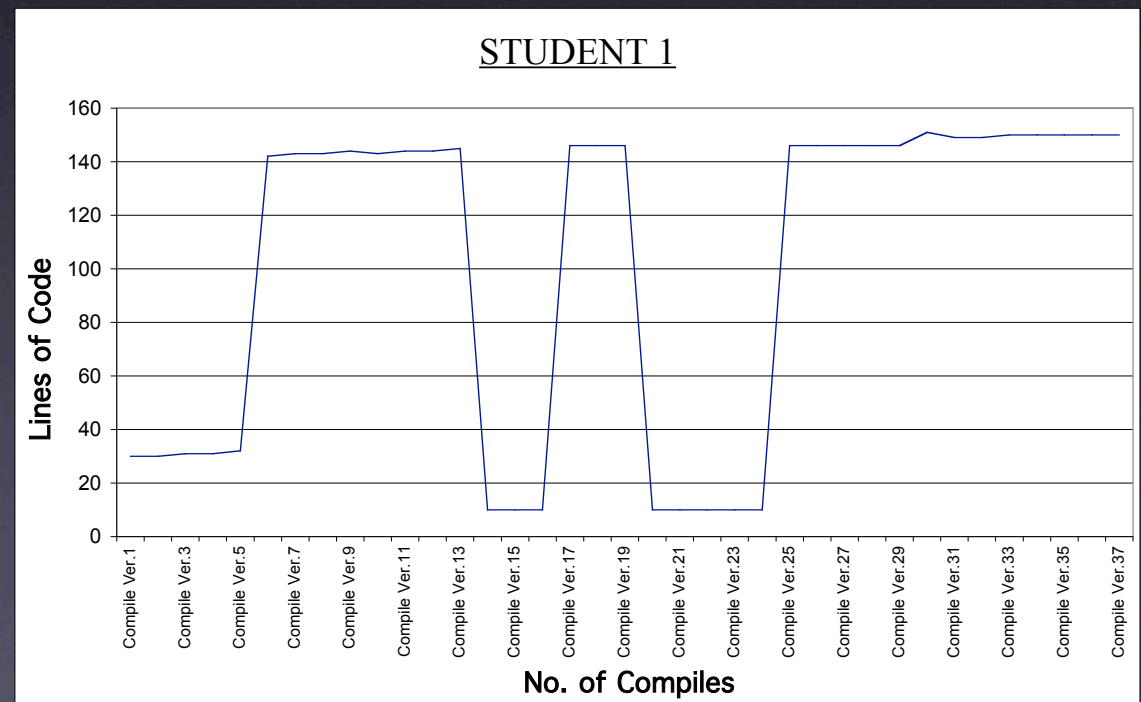
average compiles: 11 to 20
extreme: 72 in 90 minutes



differentials in constructs

- changes in expressions
- LOC

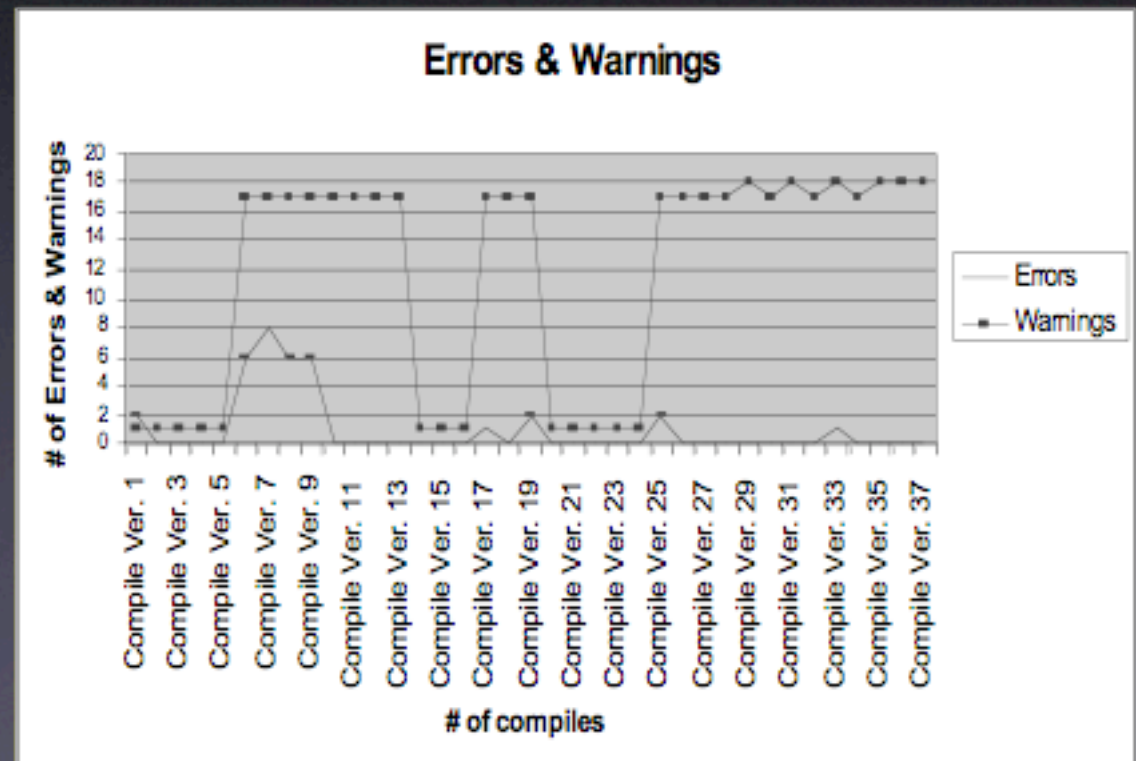
changes in LOC of a student



compiler output

- difference between compiler output between CTSs
- shows where students make mistakes

errors and warning though 37
compiles of a student



construct diffs and compiler output

- relationships between CTSs and compiler output
- particular compile error can be mapped onto a change in code

ontology

- record interactions between user and IDE
 - compiled, debugged, ran
- ▶ allows us to recognize SRL patterns

data analysis

- using SRL patterns
- CTS analysis results
- ▶ opportunities for user interaction initiation
 - try to find a place where user has finished a task or is unsure how to continue
- ★ students stop to verify code they've written (compilation)

current work

- MI-EDNA
- Learning Data Structures with Java

future work

- determine the types of interactions that are suited to help users in this domain
- implement IDE that allows us to observe user interactions
- implement IDE that allows us to initiate MI interactions

thanks

Research funded by NSERC (Lornet) and SSHRC-INE (Learning Kit)