Programming by example: efficient, but not “helpful”

Drew Goldman[2], Mark Santolucito[1],
Allyson Weseley[2], Ruzica Piskac[1]

Yale University[1], Roslyn High School[2]

May 7th 2019, Bertinoro, MOSCA 2019

About these slides and this work

- Presented at PLATEAU'18 by Drew Goldman, a high school student
- The paper is available at link
- A "younger" version of this talk can be found at
  https://www.youtube.com/watch?v=75KemPBNh2c

Scripting can be difficult

Print all .pdfs in a directory and subdirectories

e.g. print → OOPSLA18/main.pdf , PLATEAU18/figs/graph.pdf
no print → OOPSLA18/main.tex ,

```
$> find . -type f -name '*.pdf' | xargs print {} \;
```

Scripting can be really difficult

Construct links for a list of files:

e.g. file.jpg → file

```
$> sed/\(^[a-zA-Z0-9]+\)\.\([a-z]+\)/\<a href\=\"\1\.\2\"\>\1\<\/a\>/g
```

Programming by Example (PBE)

Provide examples of intended functionality with **StriSynth**

PARTITION:

Valid → OOPSLA18/main.pdf , PLATEAU18/figs/graph.pdf

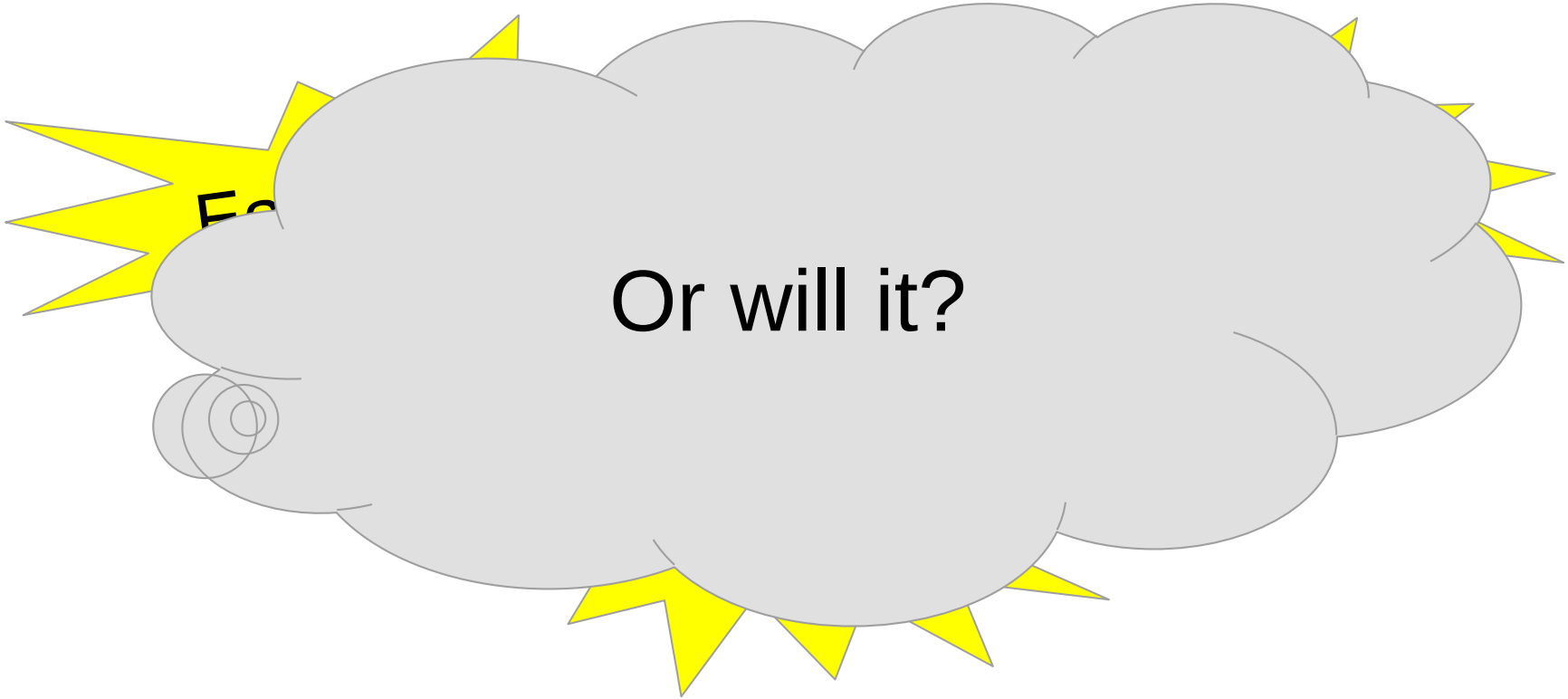
Invalid → OOPSLA18/main.tex

TRANSFORM

file.jpg → file

StriSynth: Synthesis for Live Programming, ICSE 2015, Sumit Gulwani, Mikael Mayer, Filip Niksic, and Ruzica Piskac

PBE will solve all your problems



Goal of the study

Traditionally:
Is StriSynth (and PBE for scripting) more efficient than PowerShell for real world users?

Additionally:
Is StriSynth (and PBE for scripting) a tool real world users want to use?

Are these two questions different?

Study Design

1) A tutorial on both PowerShell and StriSynth that introduced the paradigm and syntax.

2) Complete three scripting tasks in PowerShell:
    a) Extract filenames from a directory listing
    b) Move files with *.png to imgs/
    c) Printing pdfs from a list of various file types

3) Complete the same three scripting tasks in StriSynth.

4) A post-study survey.

Study Design

Run in-person for every one of the 27 participants

Participants used our laptop for a consistent development environment/experience

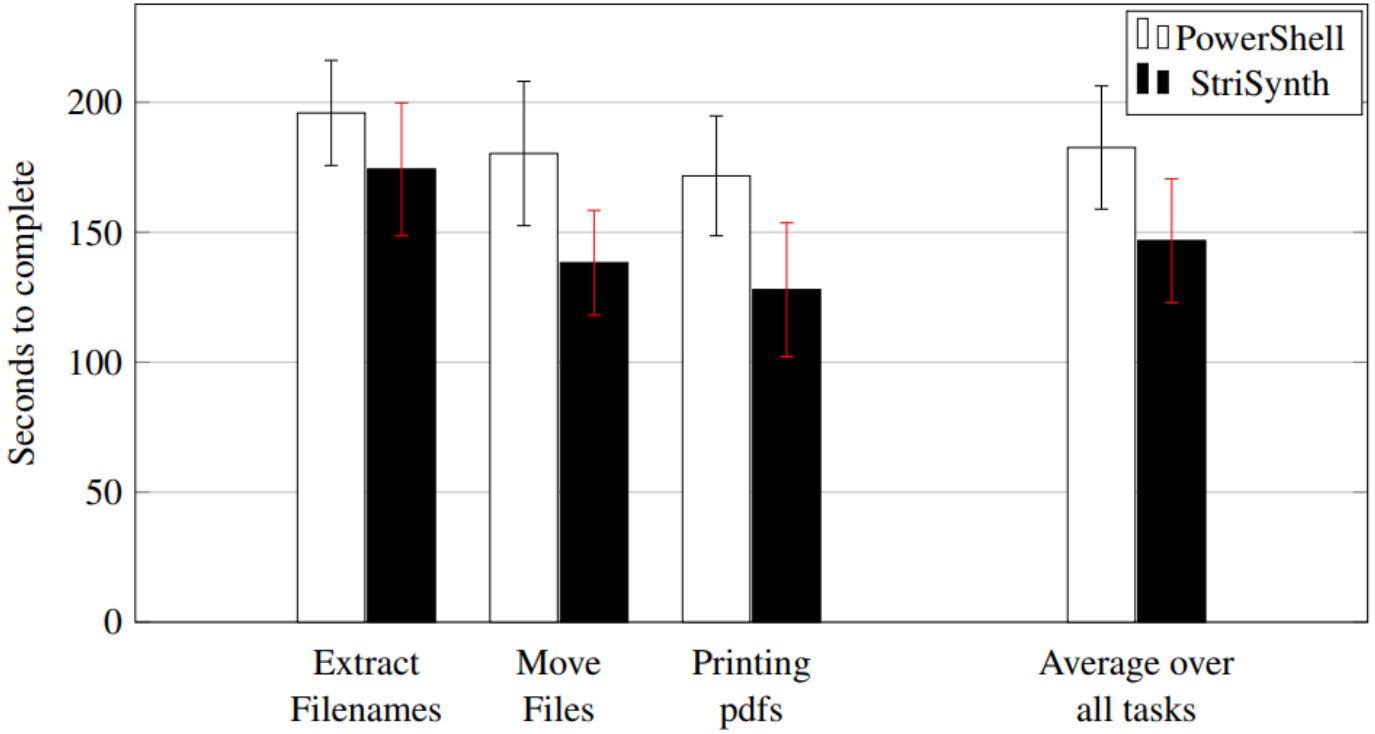
The study lasted about 50 minutes for each participant.

Used randomized condition order:

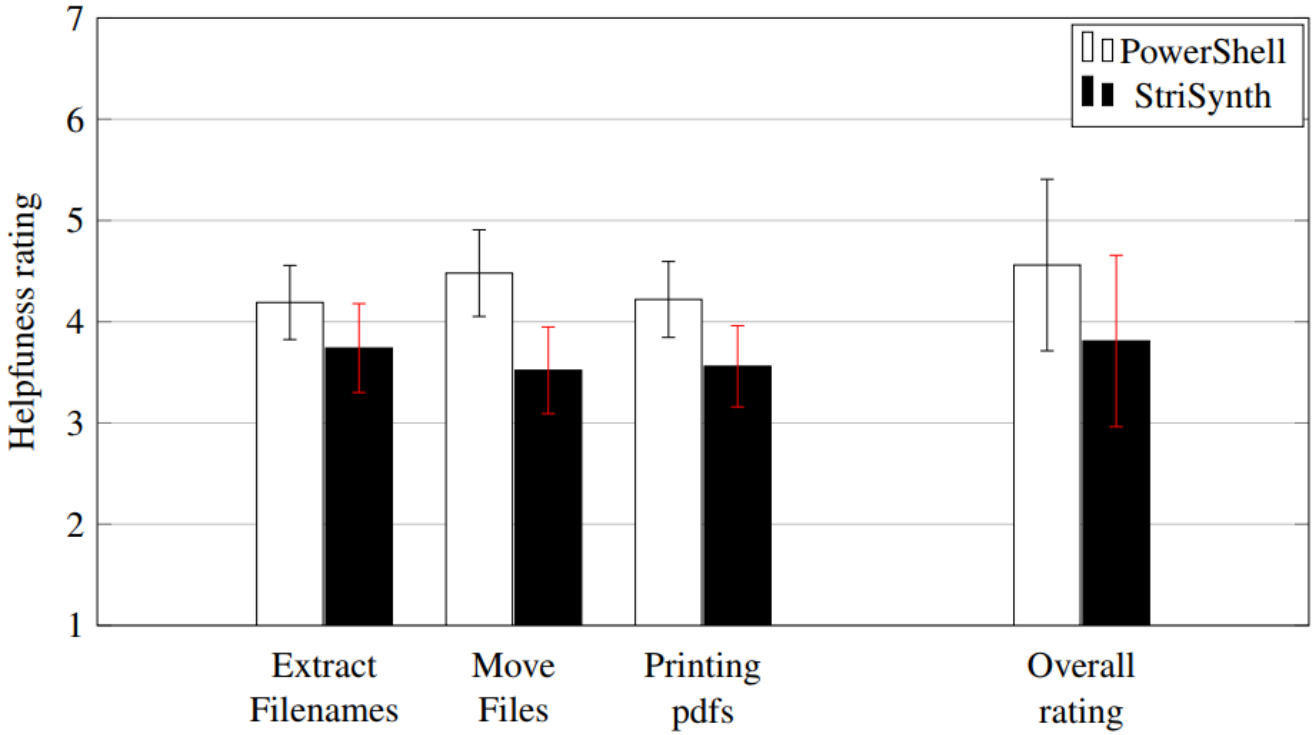
- Group A (N=12) used PowerShell (StriSynth A) first
- Group B (N=15) used StriSynth (StriSynth B) first

All materials available online: https://github.com/santolucito/StriSynthStudy

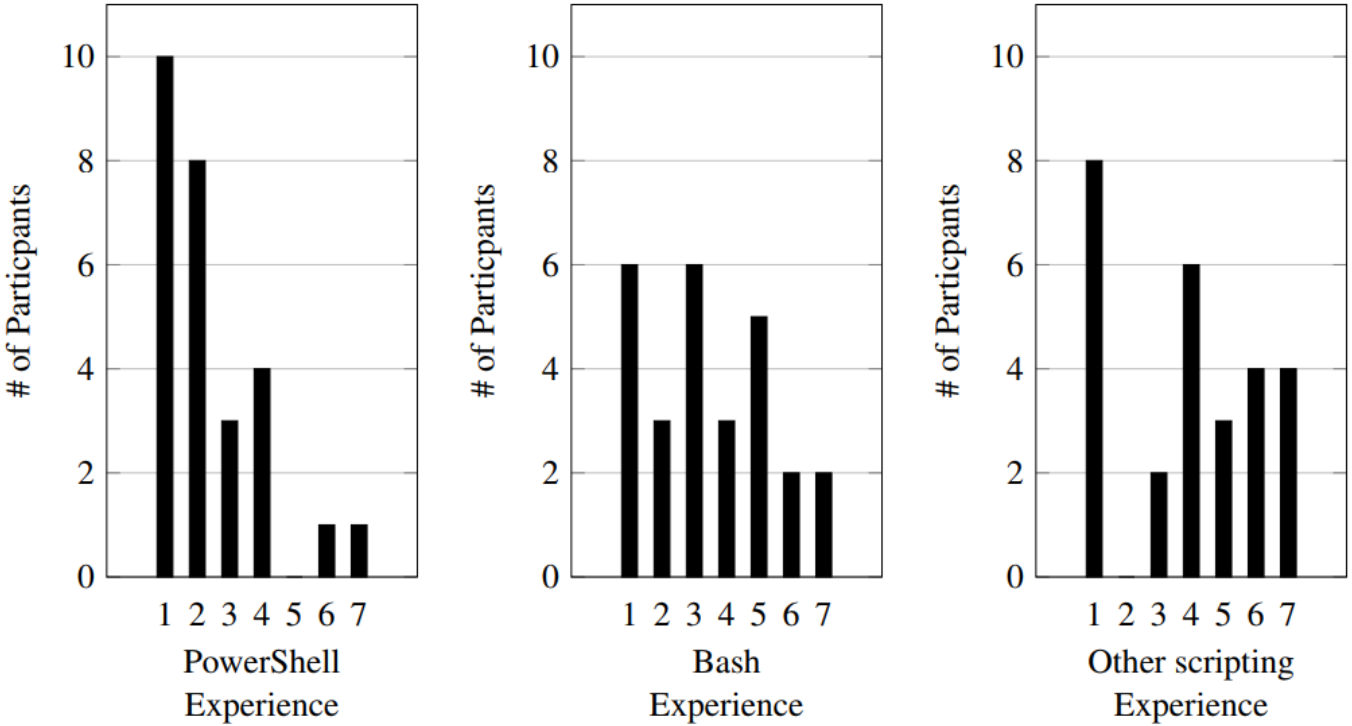
PBE is faster than scripting



PBE is less “helpful” than scripting



Participants were real world users

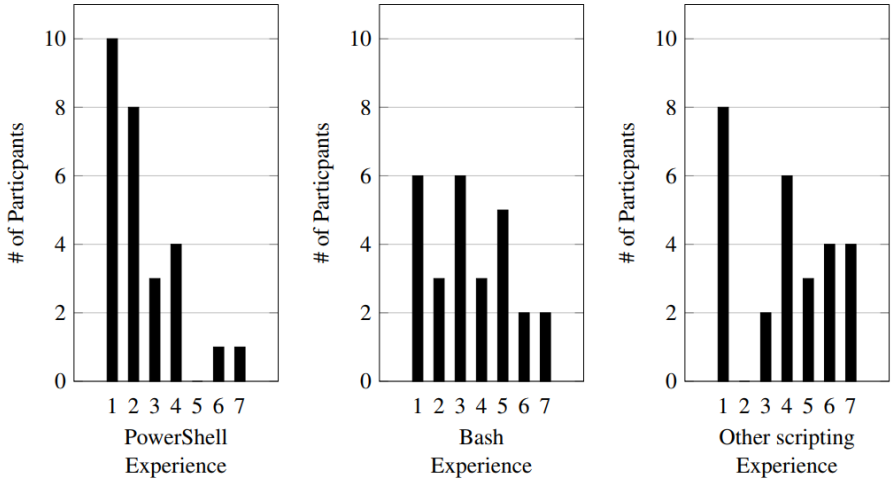


Participants were real world users

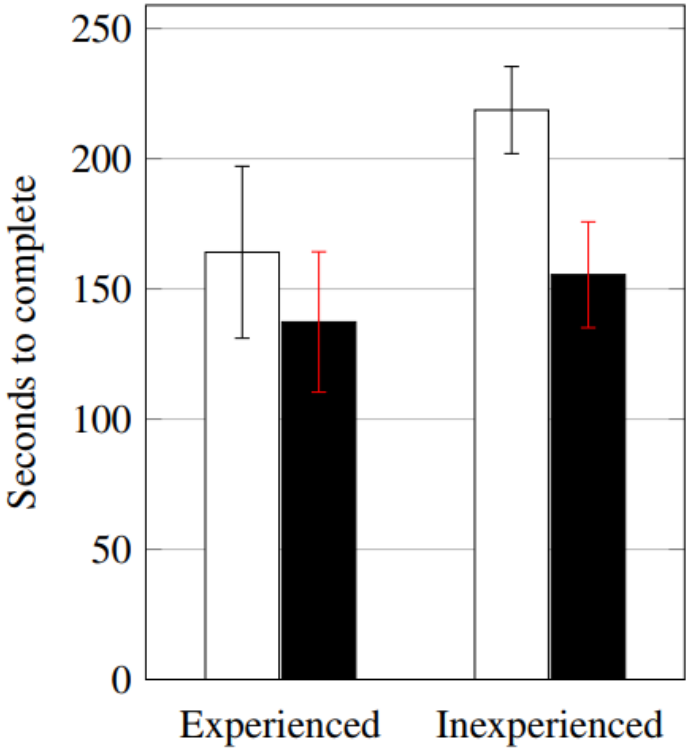
Despite some prior experience with PowerShell, StriSynth was still more efficient.

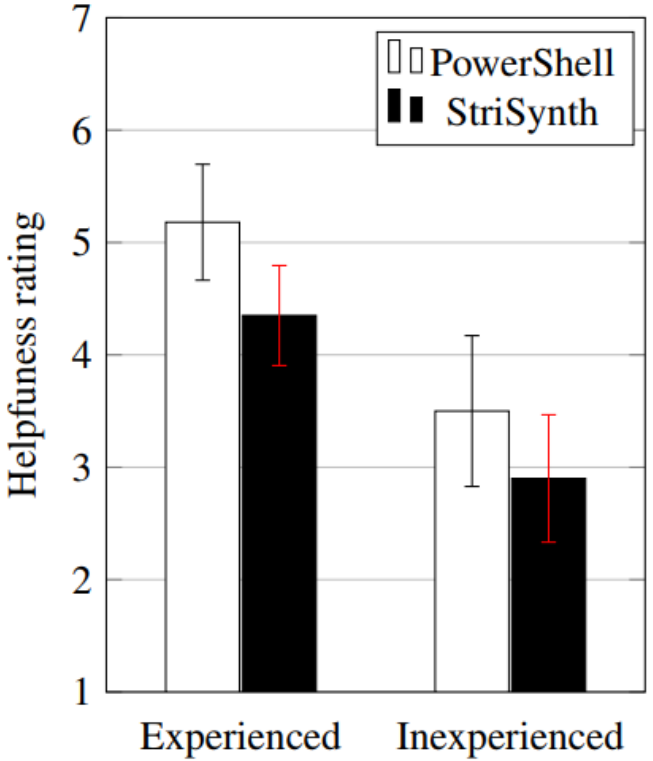
This distribution is very different when participants are PhD students.

Could this prior experience have been a factor in “helpfulness”?



Similar results across skill level





How are efficiency and helpfulness different?

- The “interface” of StriSynth
  - Similar to PowerShell (command line tool)
- Trust in the result
  - StriSynth generates an English explanation after synthesis
- Perceived lack of control
- Programmers enjoy programming

Quick tips for synthesis user studies

Measure more rather than less: Running a study has a high fixed cost, but extending each participant's study time is relatively cheap. Ask users to complete a survey with more questions than you think you need.

Need-to-please bias: Users are biased towards the researcher's tool. Present both the synthesis tool and the "traditional" approach as your own tools in the study.

Selection bias: PhD students (usually) are not the target user group - reach out to companies to recruit the "right kind" of user.

Moving forward

Further studies

- What did users mean by “helpfulness”?
- How does this transfer to other domains of PBE and program synthesis?

Generally, what benchmarks should we be using?

- Speed of synthesis algorithm
- Coverage of benchmark problems
- Speed of users to complete tasks
- Preference, helpfulness, something else