

UPIC A Problem-solving Framework:

Understand, Plan, Implement, and Correctness/Debugging

Sadhana Suryadevara and Kristin Stephens-Martinez

Introduction

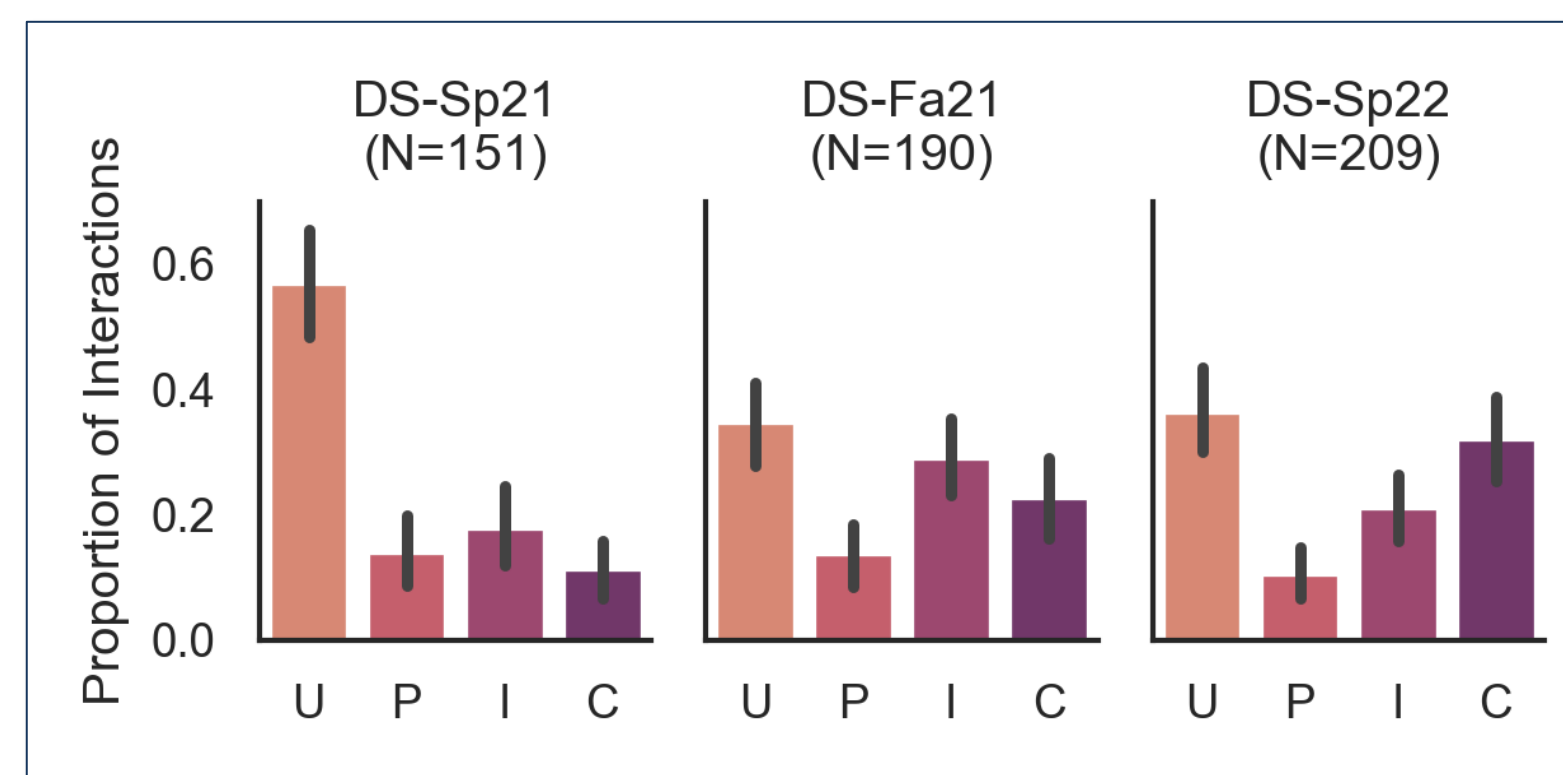
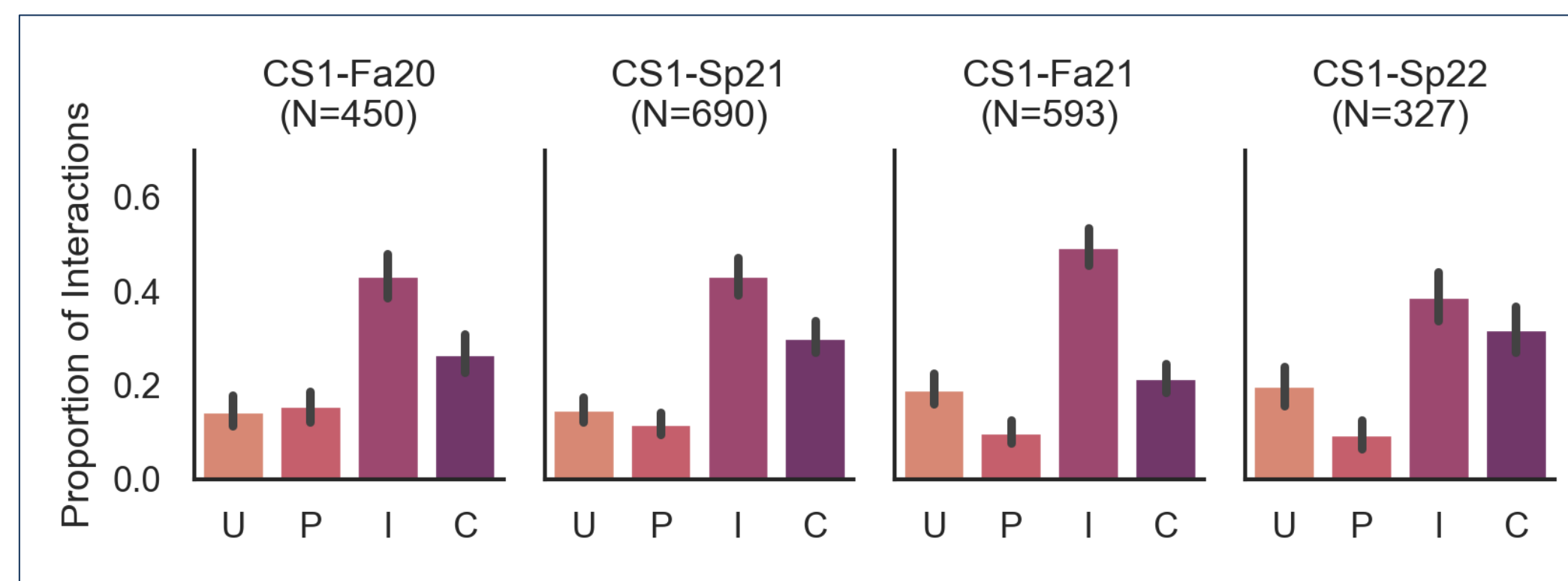
- **Problem:** Where do students get stuck?
- **Goal:** Understand when students seek help in office hours (OH)
- Courses teach **different** problem-solving processes as scaffolding
- To **compare** across courses, we need a common language around problem-solving
- UPIC **abstracts** a problem-solving process into four phases
- We applied UPIC to survey responses students provided before joining an online office hour queue for a CS1 and intermediate data science (DS) course

Method

- For each OH interaction, students reported in the pre-survey their current UPIC phase
- The CS1 used the 7-steps terminology, a problem-solving process explicitly taught in that class (see table)
- The DS course did not have a problem-solving process, so designed options using UPIC to replace an open textbox
- Data set:
 - **CS1:** Fall 2020 (Fa20) to Spring 2022 (Sp22)
 - **DS:** Spring 2021 (Sp21) to Sp22
 - From Duke University, a medium private R1

Findings

- CS1
 - Implementation most common for 3 semesters
 - Correctness usually second most common
 - Understand and Plan least common
- DS
 - Greater variation, maybe due to autograder added in Fa21?
 - Sp21 Understand most common
 - Fa21 & Sp22 Plan least common



Would you use UPIC?

How?

What else should we look for in the data?

UPIC

1. **Understand** the problem
2. Create a **Plan**
3. **Implement** the plan
4. Verify **Correctness/debug**

CS1	U	Doing an instance of the problem (Step 1 of the 7-steps)
	P	Developing a plan to solve a problem (Steps 3 and 4 of the 7-steps)
	I	Writing the code to solve a problem (Step 5 of the 7-steps)
	C	Testing my program (Step 6 of the 7-steps)
DS	U	Understanding a problem or directions
	U	Understanding a concept from class
	P	Planning how to solve a problem before getting into the math/code details
	I	Writing the math/code details to solve a problem
	C	Validating/testing/debugging my solution

Implications

- UPIC enables aggregating different problem-solving processes for easier comparison
- Knowing the most common reason students seek help can inform TA training
- Autograders potentially influence when students seek help
- Teachers with no explicit problem-solving process could use UPIC to see where students struggle or to create a process