Exploring Novice Compilation Behavior

Matt Jadud
20060910
ICER 2006

Map

- **Background**
- Data
- Explorations
  (and things found along the way)
- Discussion

In the beginning...

- **Vincent**, Indiana University Bloomington

  Students used this web-based course management system to hand in code.

- **Behavior**
  - Some submit 5 times...
  - Some submit 50 times...

The big lie

- **Lie**: Students are just letting the compiler do their thinking for them.
- **Problem**: Dismisses the power of modern computers (“cycles to burn”)
- **Truth**: *Professionals* let the compiler do their thinking for them (eg. Eclipse, test-first, etc.)

Back in my day...

- **Lie**: Programmers made fewer mistakes in the days of the punch.
- **Truth**: If you remember writing perfect code, either you're *wrong*, or you're *special*.

And, you probably walked uphill both ways to school, too.

We make mistakes

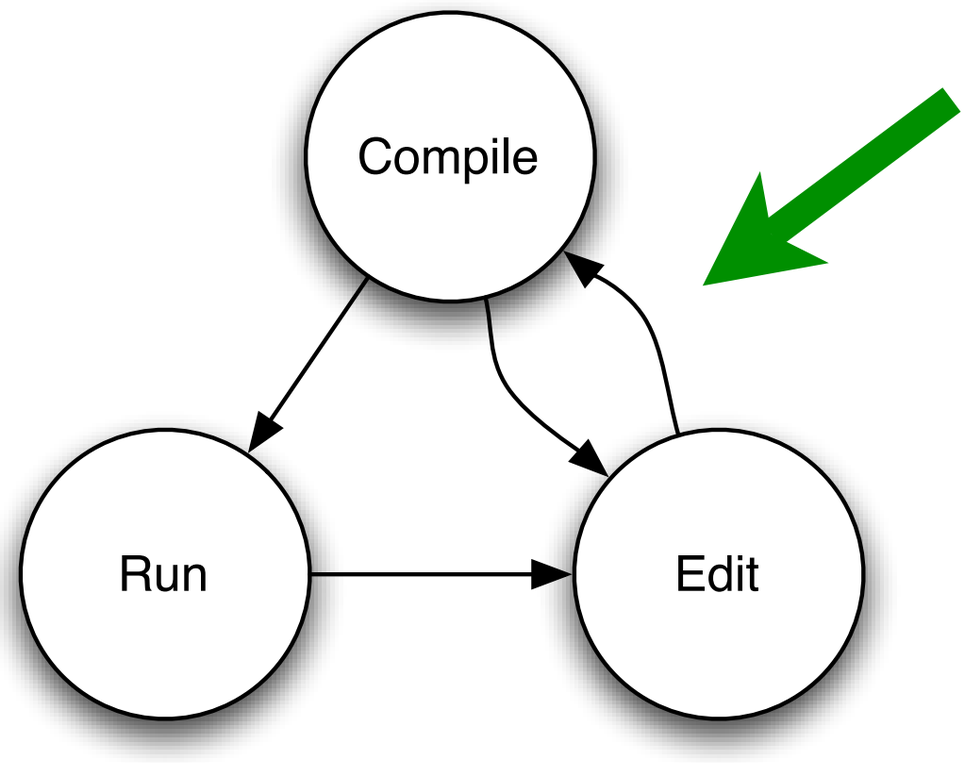
- **CoRC**: PL/C, 4000 runs, req'd 2 runs/prog to "achieve acceptable operation"
- **Ditran**: 36% of all submissions contained syntax errors (~ 4/prog)
- **Pilot**: 51% of all student compilations ended in error
- **Study**: same

Map

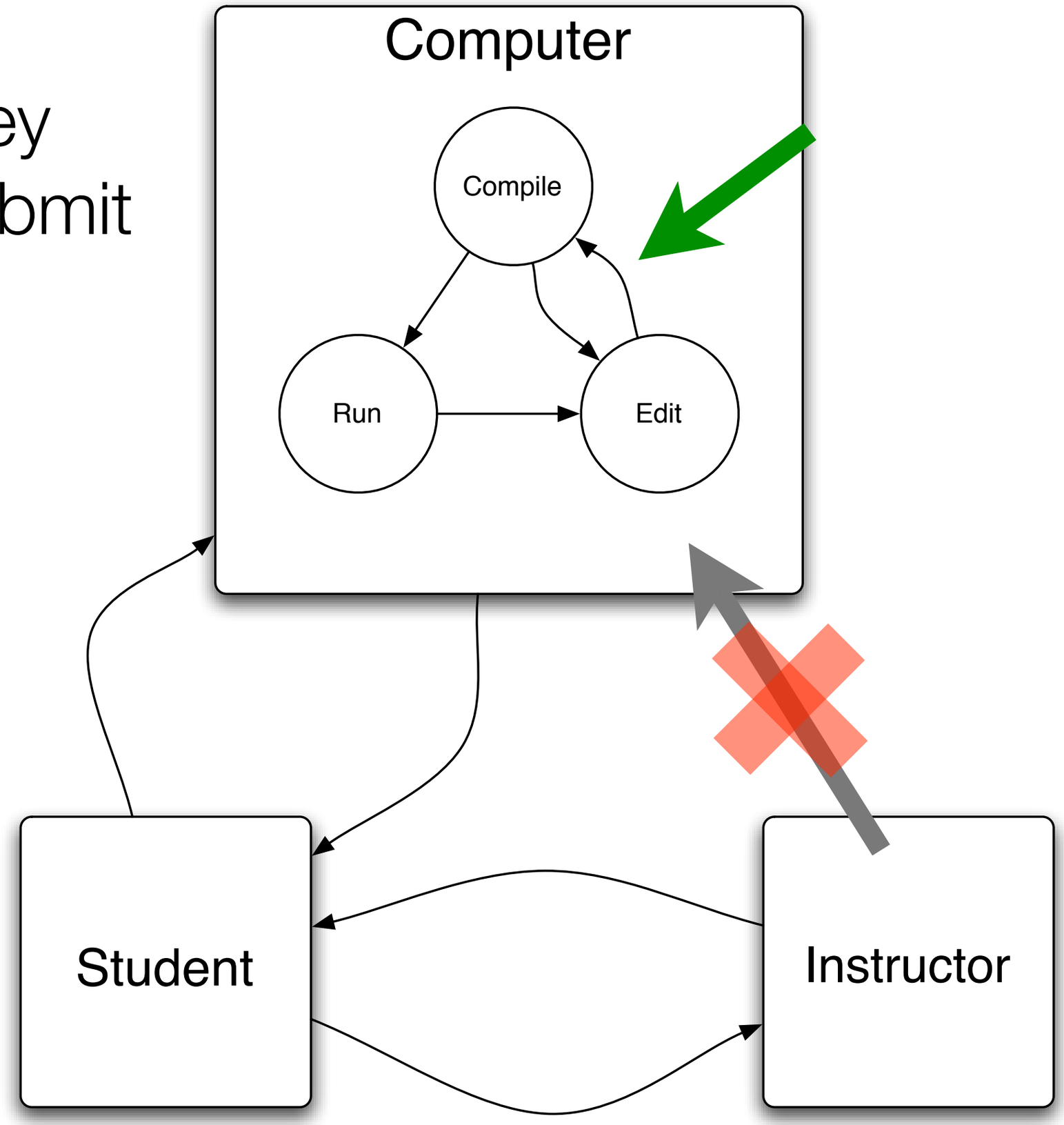
- Background
- **Questions**
- Explorations
  (and things found along the way)
- Discussion

(Deceivingly Hard) Question

- What are they *doing* when they recompile/resubmit so often?

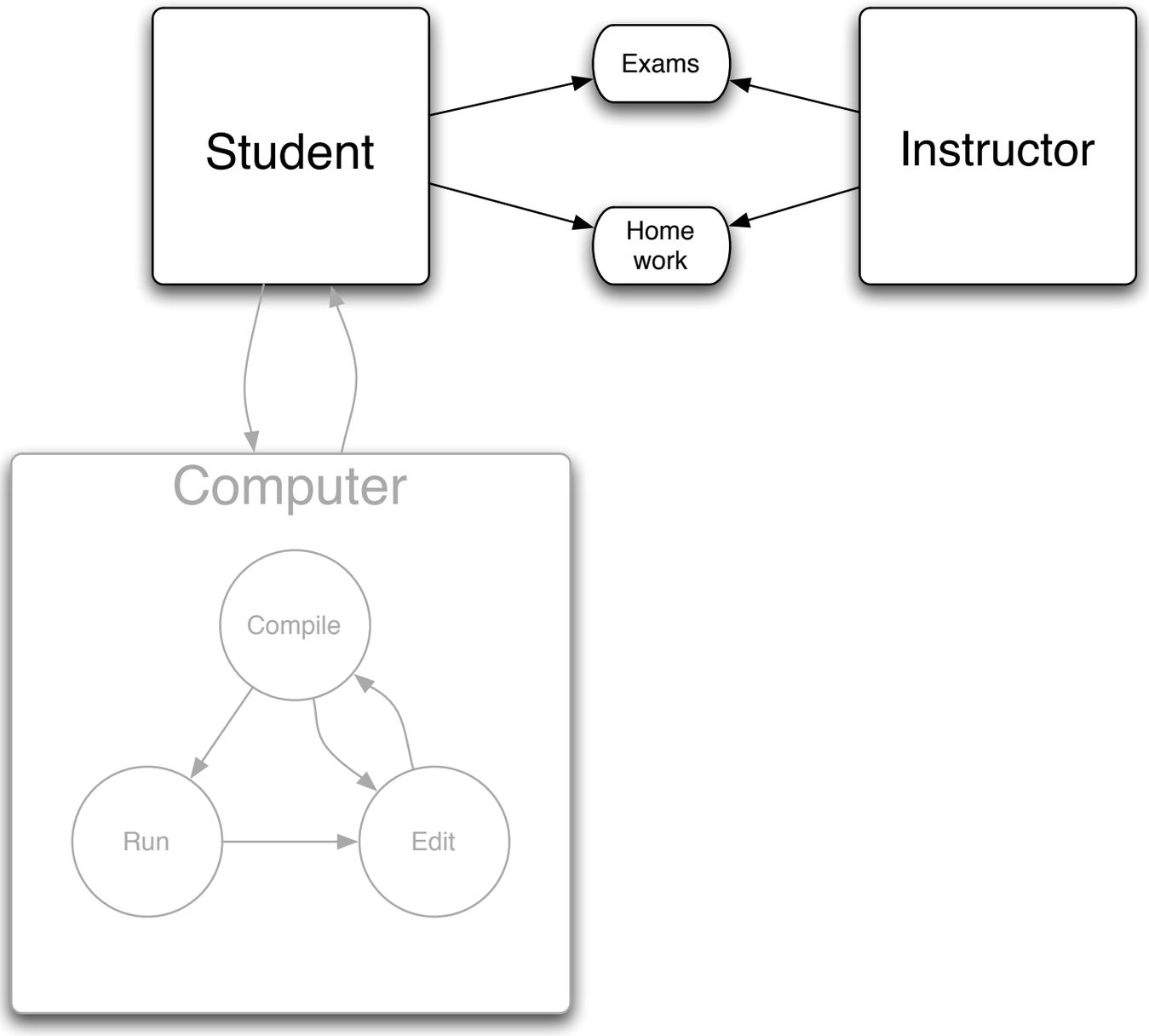


What are they *doing* when they recompile/resubmit so often?



(Obviously Hard) Question

- Whatever it is they are doing, does it relate at all to our assessments of their abilities?



Over-arching Question

- Is their programming behavior getting in the way of good/effective learning?
- Can it be shaped, or otherwise improved?

Map

- Background
- Questions
- **Explorations**
  (and things found along the way)
- Discussion

Data

- Captured a copy of student programs every time they pressed the “compile” button.
  - 2 years,120 students, 2000 programming sessions, 42,000 snapshots

Data Archeology

- **Search**: Spend months figuring out where to do the digging.
- **Tools**: If you have complex data, don't go looking for simple answers.
- **Suggestion**: Dig long and deep through your data; all good quantitative inquiries start with an extensive qualitative search.

Browsing code

- First things first: if you're studying compilation deltas, start reading.
- When you see the "big picture," and can characterize general behaviors in your population, stop.
- *Haven't really reached step two...*

Click me! Click me!

Work smart, not hard

- When you're faced with thousands of sessions, and tens of thousands of compilations, **get help**.
- If you can't get the help you need, **automate** or **abrogate**.

Focus

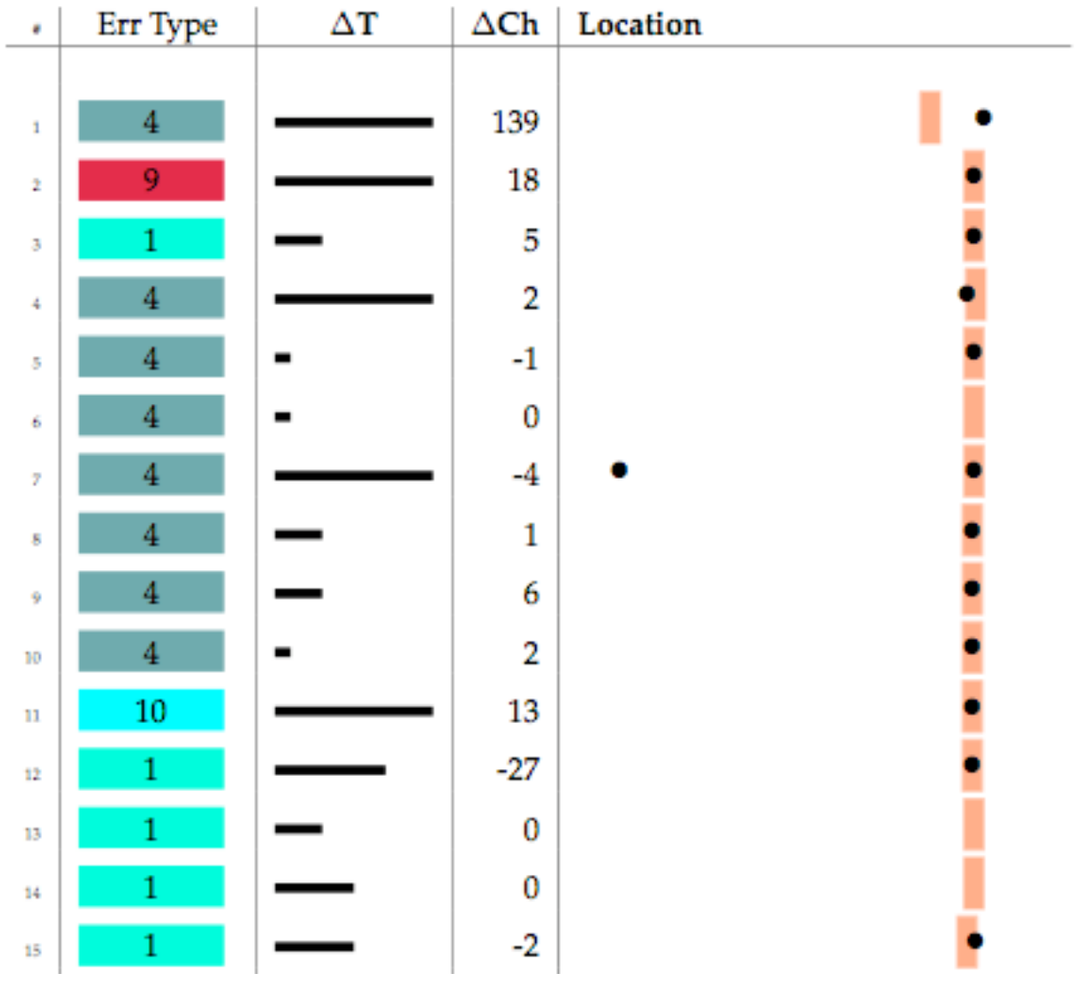
- What seems important?
  - Error type and location
  - Repetition of error
- These things, in combination, highlight problem spots in a session.

Click me! Click me!

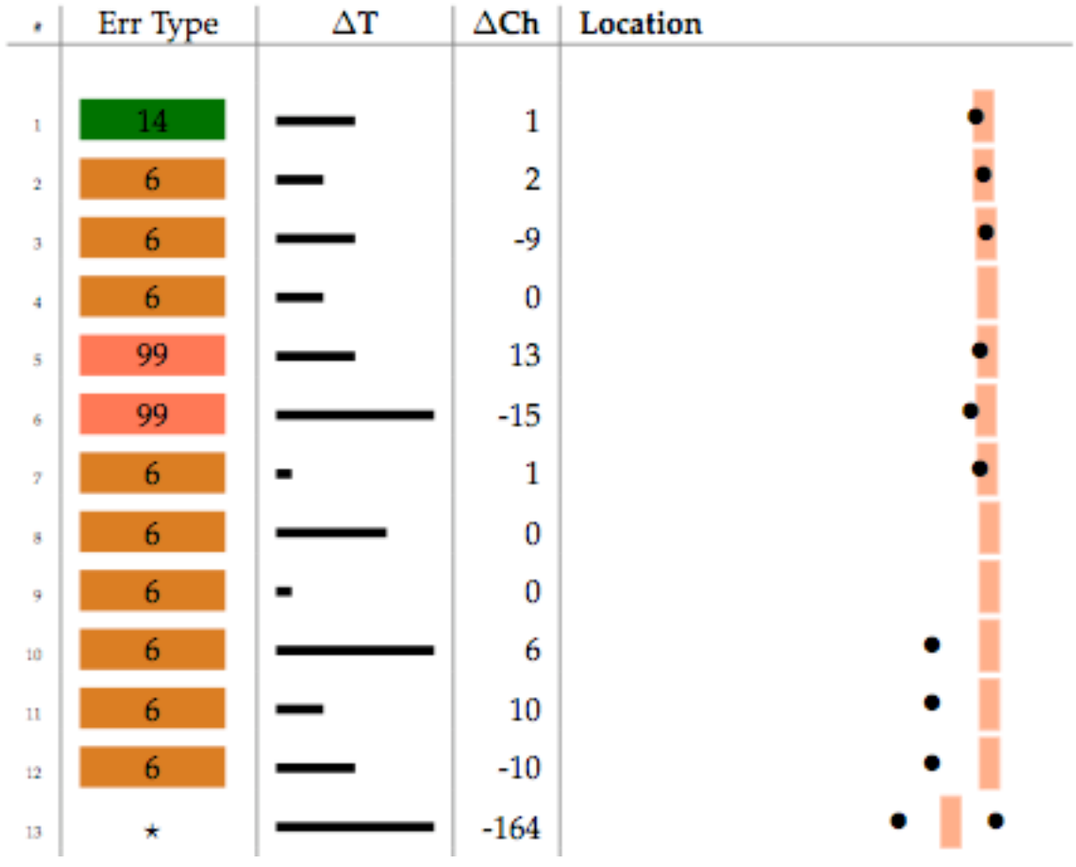
What are they doing?

- With a high-level view of a session, we can now scan through hundreds of compilation events and dozens of sessions in an hour.
- We still don’t know exactly what they’re doing, but we believe there are critical differences.

From the “bad”…

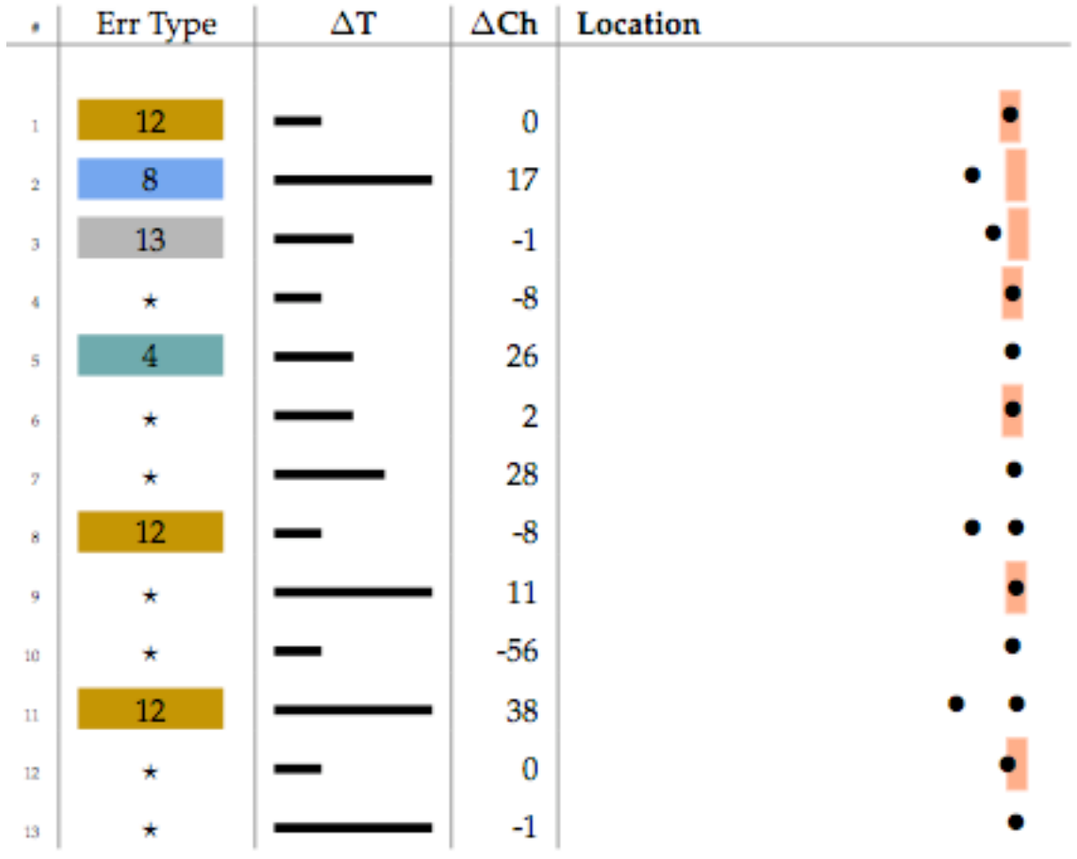


#	Err Type	ΔT	ΔCh	Location
1	4		139	
2	9		18	
3	1		5	
4	4		2	
5	4		-1	
6	4		0	
7	4		-4	
8	4		1	
9	4		6	
10	4		2	
11	10		13	
12	1		-27	
13	1		0	
14	1		0	
15	1		-2	

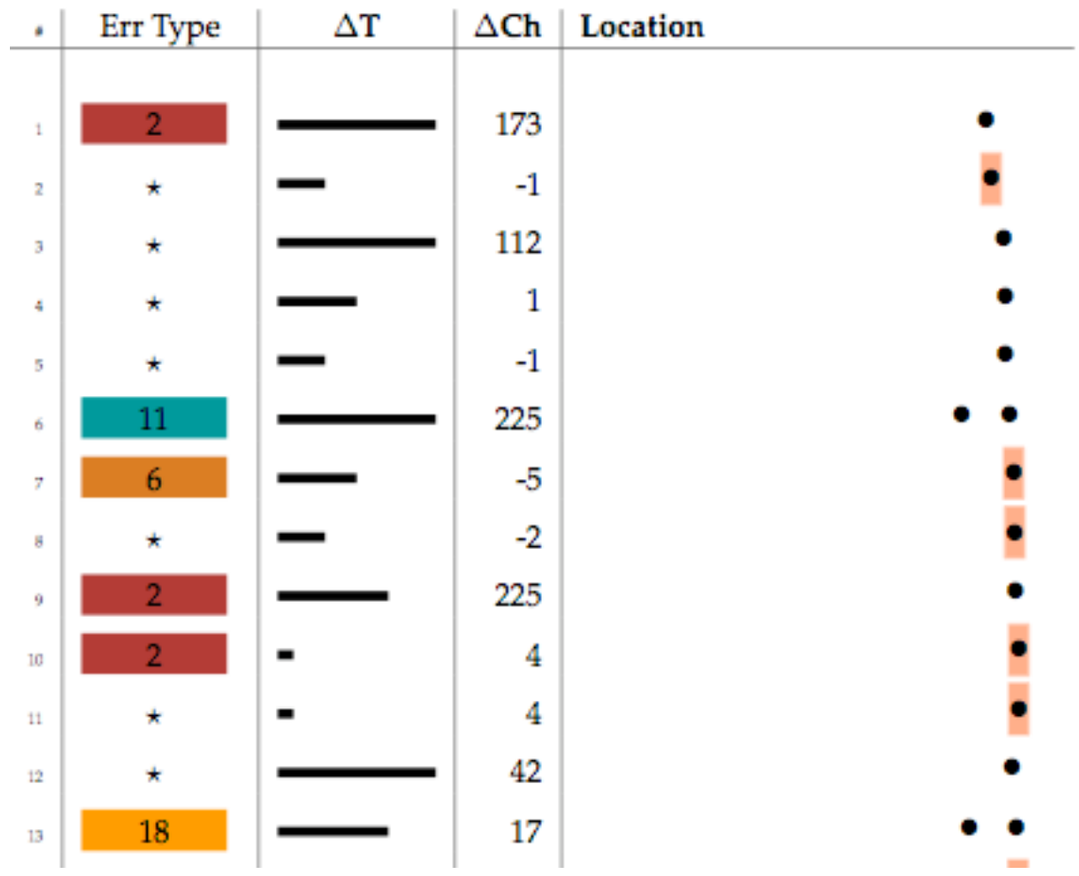


#	Err Type	ΔT	ΔCh	Location
1	14		1	
2	6		2	
3	6		-9	
4	6		0	
5	99		13	
6	99		-15	
7	6		1	
8	6		0	
9	6		0	
10	6		6	
11	6		10	
12	6		-10	
13	⋆		-164	

… to the “good”



#	Err Type	ΔT	ΔCh	Location
1	12		0	
2	8		17	
3	13		-1	
4	⋆		-8	
5	4		26	
6	⋆		2	
7	⋆		28	
8	12		-8	
9	⋆		11	
10	⋆		-56	
11	12		38	
12	⋆		0	
13	⋆		-1	



#	Err Type	ΔT	ΔCh	Location
1	2		173	
2	⋆		-1	
3	⋆		112	
4	⋆		1	
5	⋆		-1	
6	11		225	
7	6		-5	
8	⋆		-2	
9	2		225	
10	2		4	
11	⋆		4	
12	⋆		42	
13	18		17	

Oooh... numbers

- All of the things we're looking at now (error type, error location, repetition of errors) are quantifiable.
- What happens if we assign a "penalty" to each of the behaviors we've just seen?

Turn the crank

#	Err Type	ΔT	ΔCh	Location	SCORE
1	4		2	0	
2	⋆		13		0
3	1	3	-27	8	0
4	1		-2		11

$$\frac{0 + 0 + 11}{11 * 2} = \frac{11}{22} = .5$$

Naming

- Naming is *hard*
- Naming *shapes thought*
  - Eg. Is it
    - ... a *rate*?
    - ... a *measure*?
    - ... a *count*?

THE

JAR OF CONSTANT INFUSIONS OF GOODNESS

Working with Sally

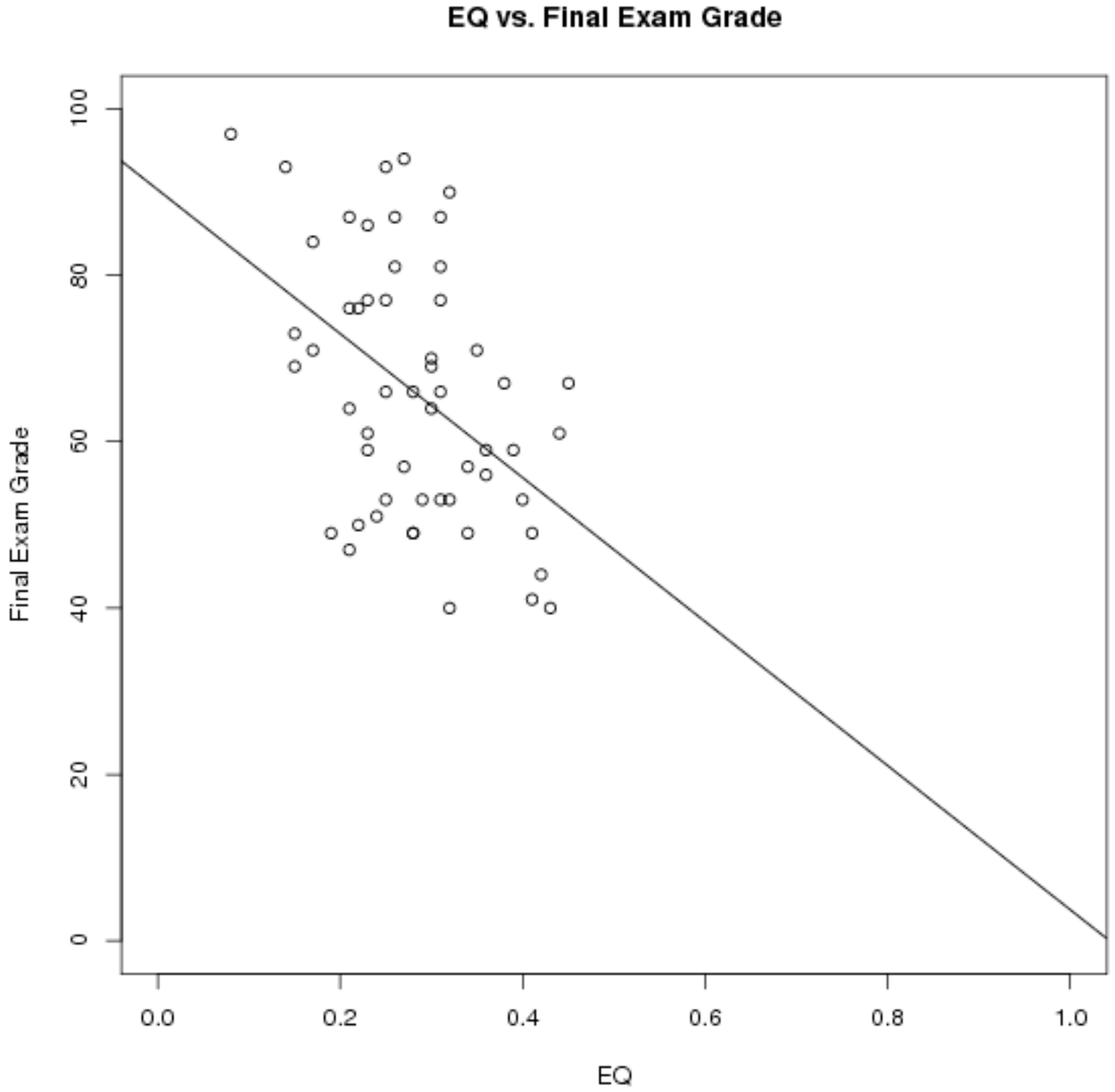
- Working with Sally is *hard*
- Working with Sally *shapes thought*

Or...

We might call this the **error quotient**.

The EQ

- Sessions where a student wrestles with one error (or several errors in the same place) score high.
- Sessions with many syntactically correct compilations score low.




```
Residual standard error: 13.91 on 53 degrees of freedom
Multiple R-Squared: 0.213,      Adjusted R-squared: 0.1982
F-statistic: 14.35 on 1 and 53 DF,  p-value: 0.0003901
```

EQ: Mostly useless?

- We probably cannot make meaningful claims (statistically) from the data we have to hand.
- But that’s OK.

Map

- Questions
- Background
- Explorations
- **Discussion**

What now?

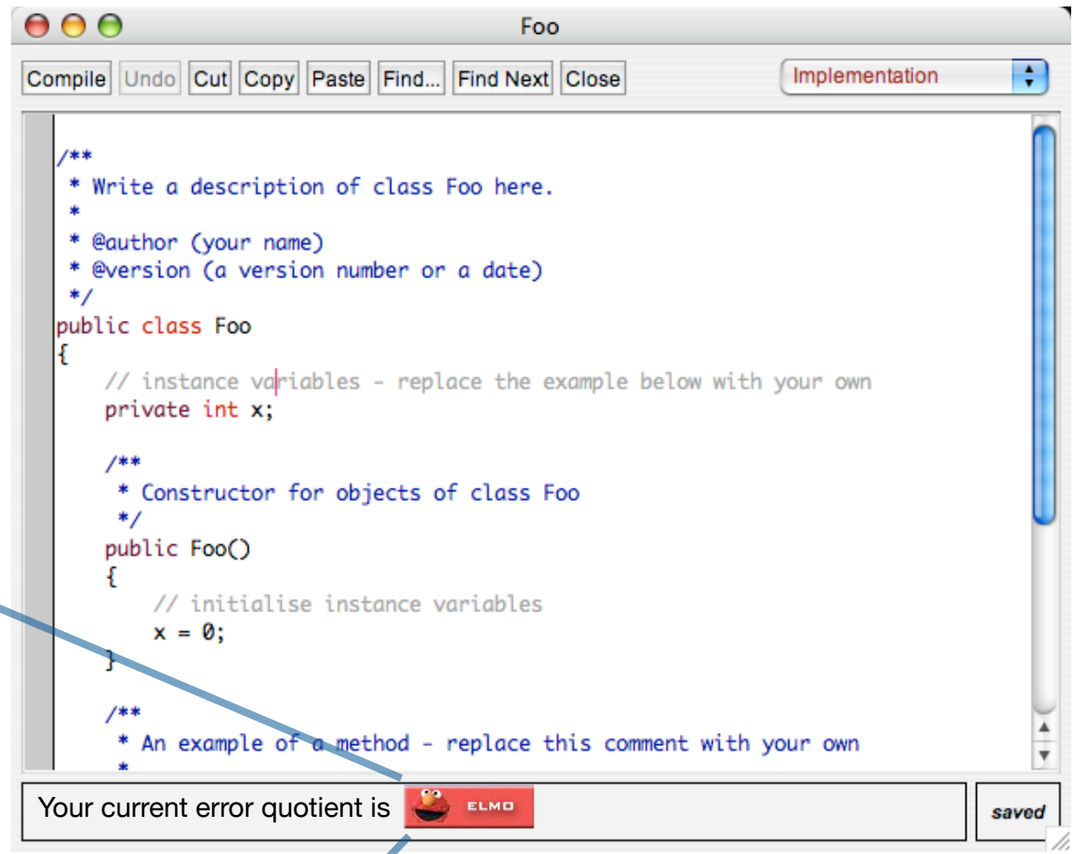
- Continue observation
  - More data may improve statistical
- Explore use of the tools in context
  - Can they help the instructor?
  - Can they help the learner?

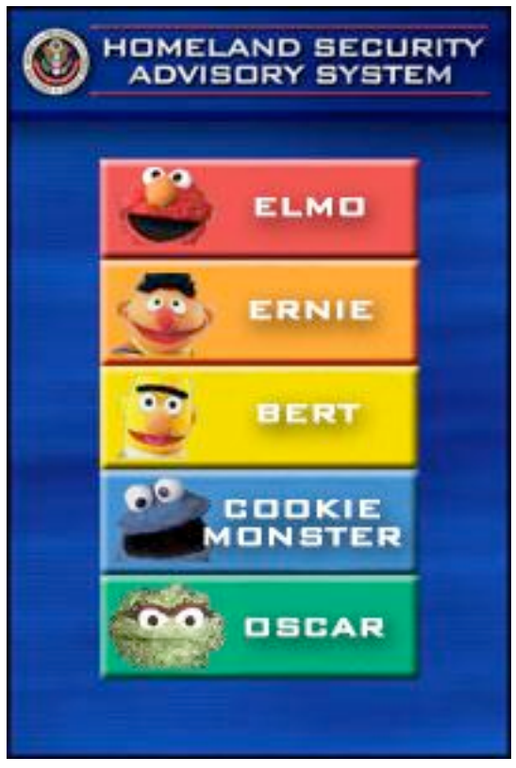
Dive in

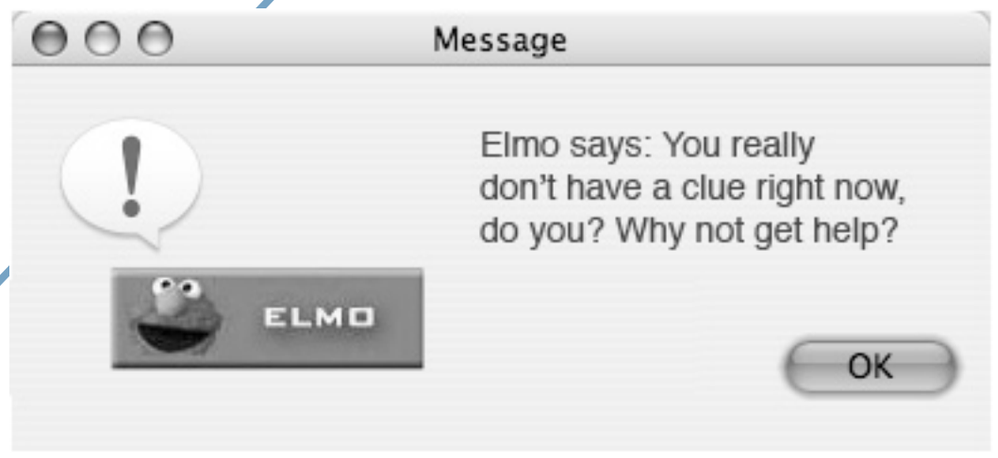
- If we think it's reasonable, start using it as a guide.
- The EQ can provide a formative, or ongoing, view of student programming that summative assessments (homeworks and exams) might not

Real-time

- The EQ can be calculated with as few as three events.







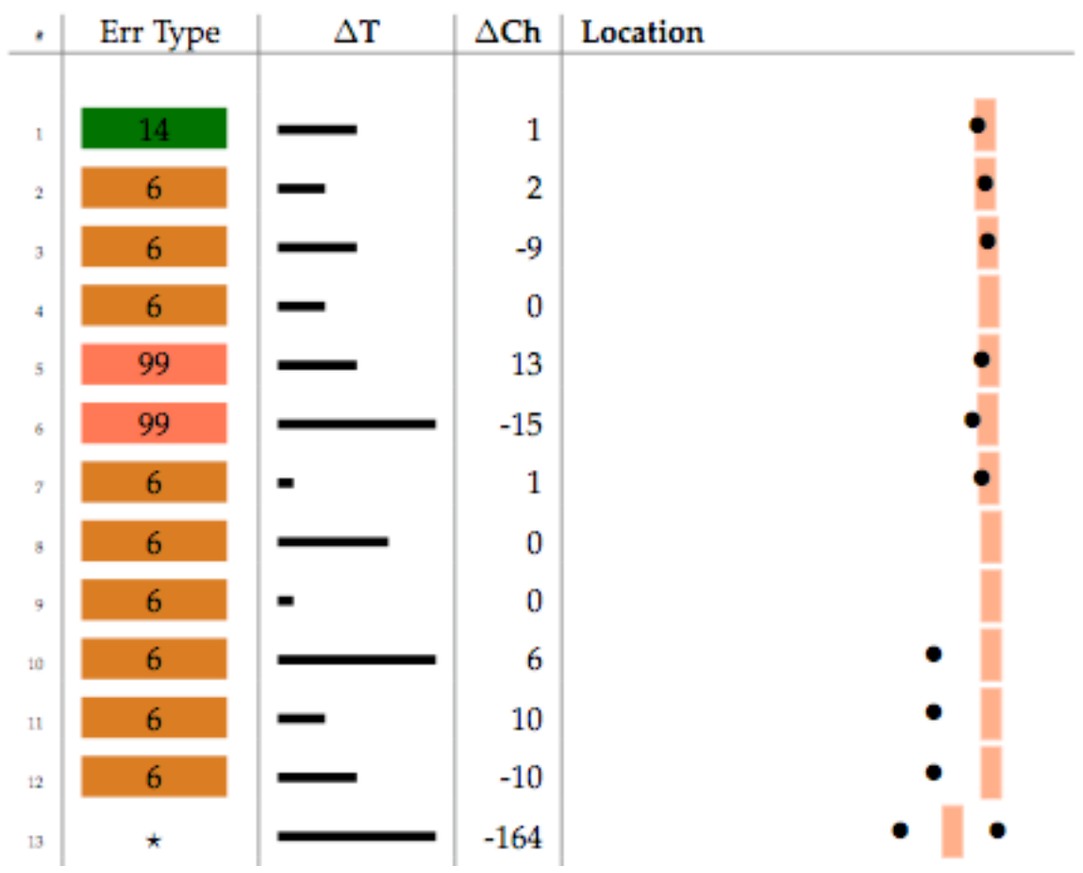
Or just-in-time

- As well as real-time feedback and enhanced help, we might provide tutorials that students can access at their leisure
- Tutorial content would be driven by real-world problems.

Click me! Click me!

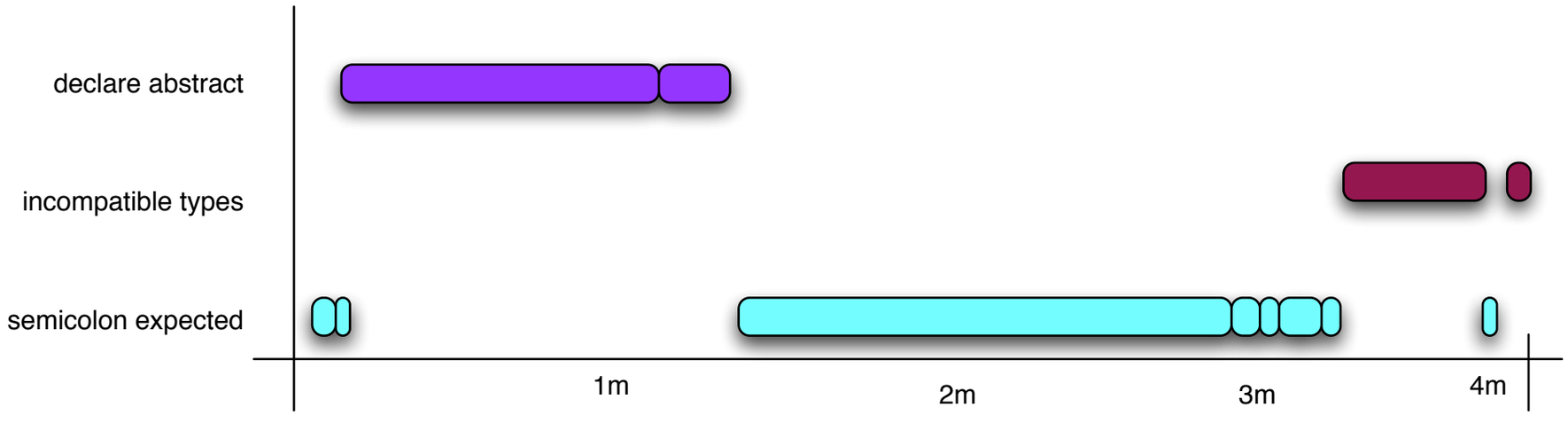
Or not so real-time

- Perhaps at the end of the day the instructor could download the sessions for some or all of the students.
- Targeted help can be offered/suggested for students who are clearly struggling with their programming.



Improving visuals

- Tabular form developed for exploring data
- Interesting work to be done improving the tools



Future work

- Bringing the EQ into play opens the door for many interesting studies regarding student interaction with code and their IDE (BlueJ).
- The EQ is, largely, language-agnostic; what about other languages and environments?

Thanks.