

Measuring Students' Performance on Programming Tasks

Tomáš Effenberger

Radek Pelánek

Masaryk University Brno, Czech Republic

Motivation

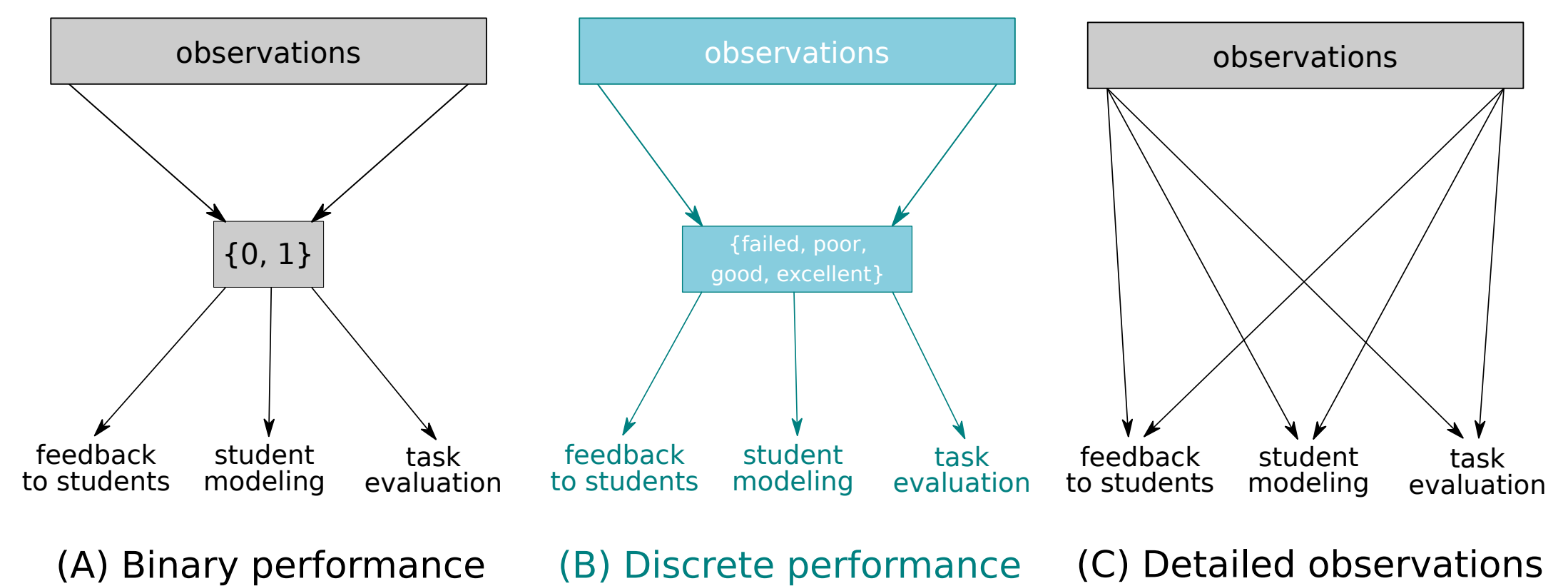
Binary success provides too little information for student and tutor modeling.

Using detailed observations hinders reuse, and complicates the development of learning systems.

Proposal: Use discrete performance measures with just a few distinct values.

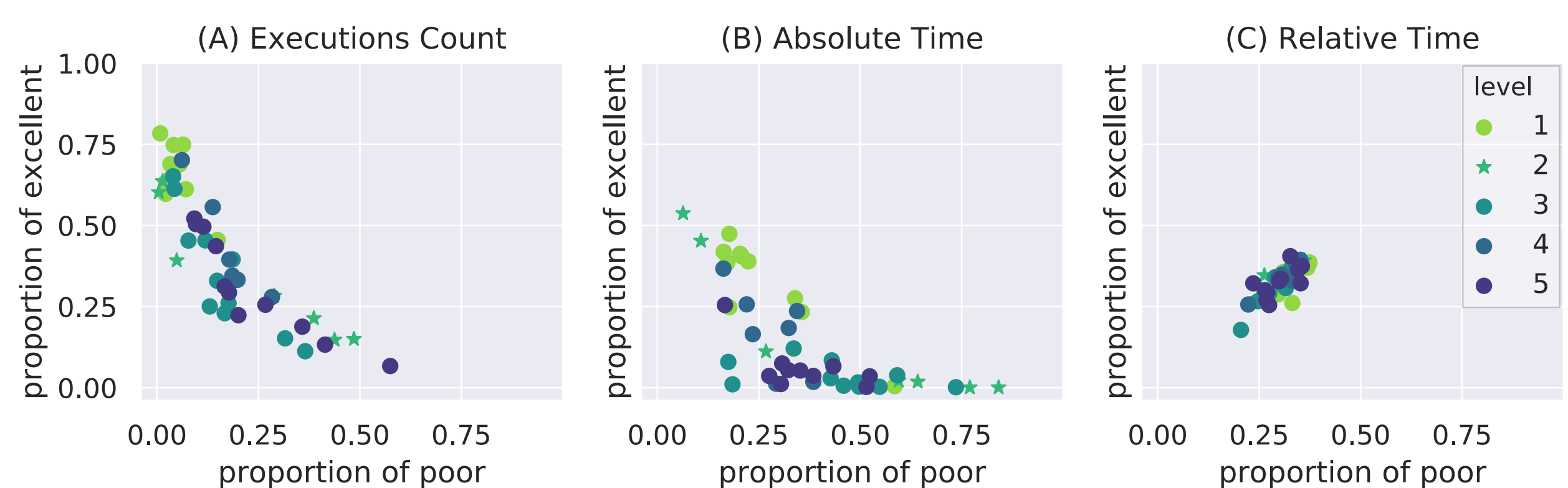
Performance levels and interpretation:

- week performance ... too difficult task
- good performance ... appropriate task
- excellent performance ... too easy task



Case study: Design of a performance measure

- adaptive learning system for introductory programming
- 85 tasks, 9 levels x 3 sublevels
- performance measures based on
 - (A) executions count
 - (B) solving time, thresholds per sublevel
 - (C) solving time, thresholds per problem



Result: The choice of performance measure matters. Different reasonable choices lead to very different measurements.

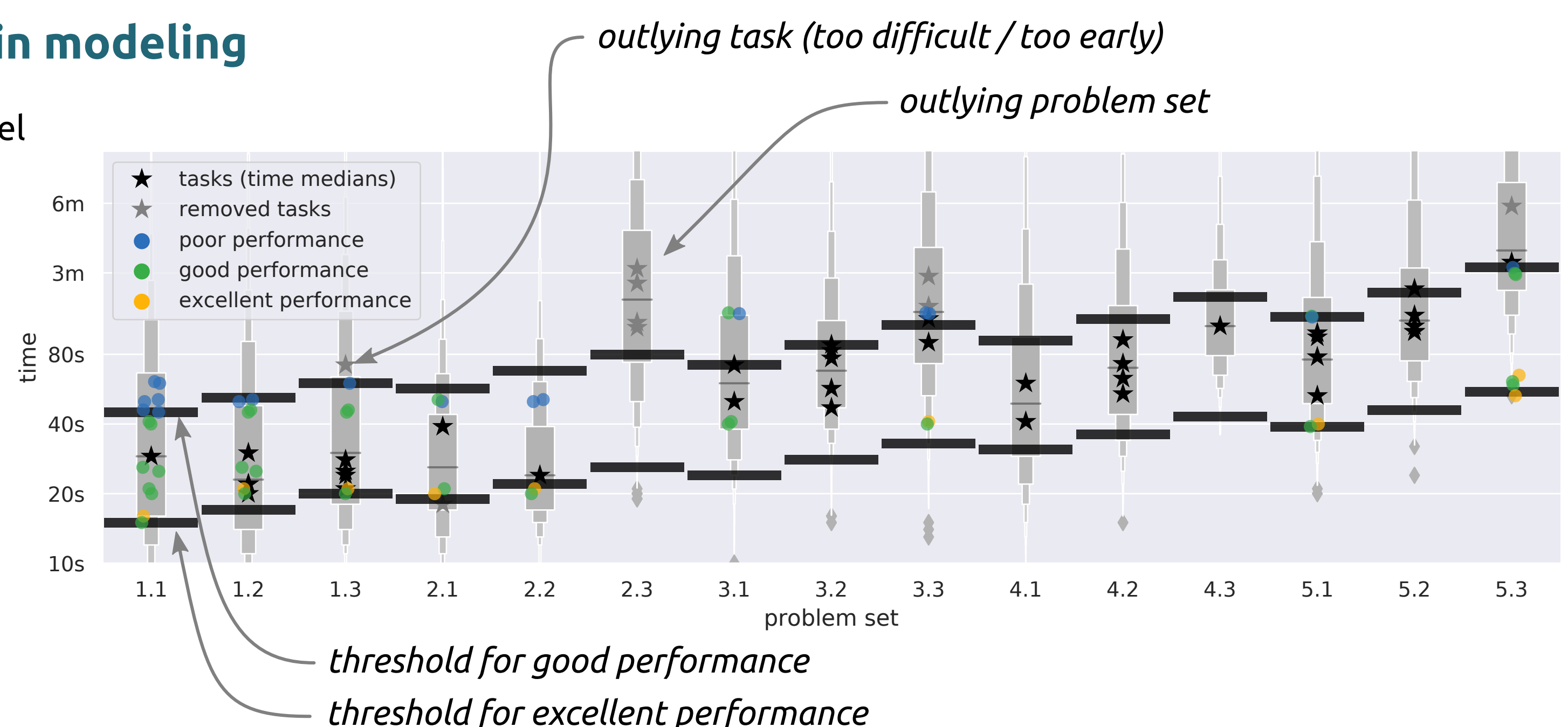
Interaction with domain modeling

Issue: Flaws in the domain model impact the accuracy of a performance measure.

Consequence: Unconstrained optimization leads to a wild performance measure.

Solution: Impose a constraint on the threshold progression (e.g., a constant increase between sublevels).

Side effect: The flaws in the domain model are detected.



Summary

Instead of using binary success, the currently prevalent choice of performance measure, we propose to use a few discrete performance levels with universal interpretation, such as failed, weak, good, and excellent.

Design of a performance should take into account possible interaction with domain modeling.