Code Snapshots Working Group Update

Presenter: Thomas W. Price
Group Leads: David Hovemeyer, Kelly Rivers
Other Contributors: Austin Cory Bart, Ge Gao, Ayaan M. Kazerouni, Brett A. Becker, Andrew Petersen, Luke Gusukuma, Stephen H. Edwards, David Babcock

Working Group Goals

Goal: Make it easier for researchers to share programming snapshot *data*, *analyses*, and *tools*.

Motivation: data can lead to insights and tools that improve student outcomes.

```
print("Hello, World!")
```

```
#include <iostream>

int main(){
    std::cout << "Hello, World!" << std::endl;
    return 0;
}
```

Challenges

Challenge: datasets collected by researchers vary greatly in programming language, program size, and which metadata was collected

Challenge: the format must be general (to handle different datasets), but also specific (to allow for meaningful analysis)

The ProgSnap2 Format

The data format

- Design goals
- Concrete representation
- Structure and semantics

Data Format: Design Goals

Desired properties:

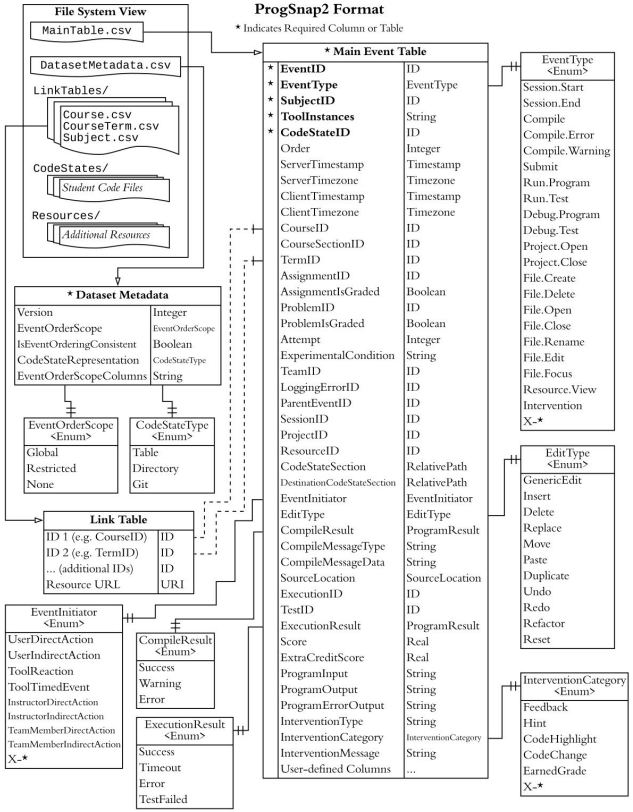
1. Capable of modeling data from diverse sources
   ⇒ working group members had significant experience with programming activity data
2. Explicitly represent what is known and unknown
   ⇒ avoid “synthesized” data values; many fields are optional
3. Easily consumed by standard tools and libraries
   ⇒ tabular data stored in CSV files

Data Format: Concrete Representation

Three kinds of files:

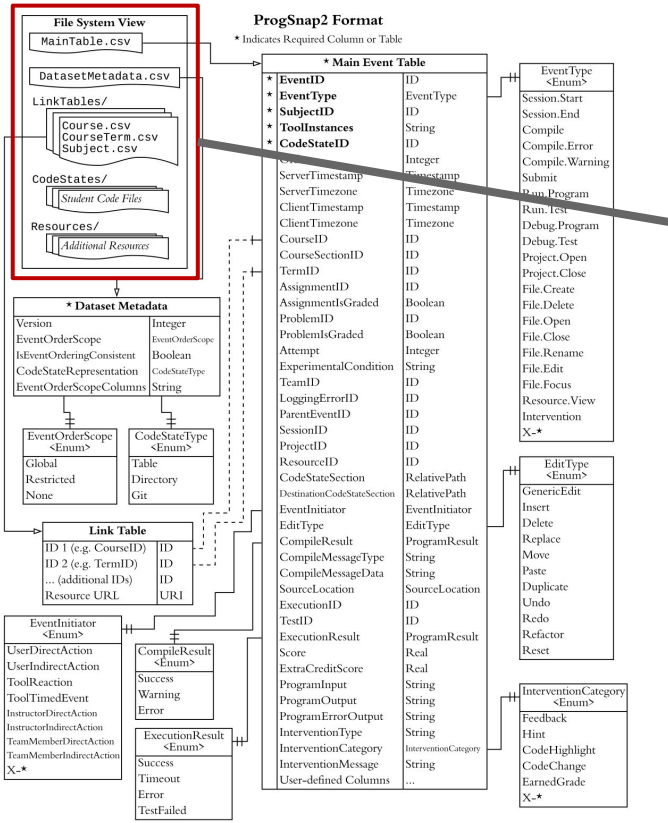
- CSV files
  ⇒ Used for all structured data
- Code states
  ⇒ Capture student code
- Resources
  ⇒ Allow associated data to be included

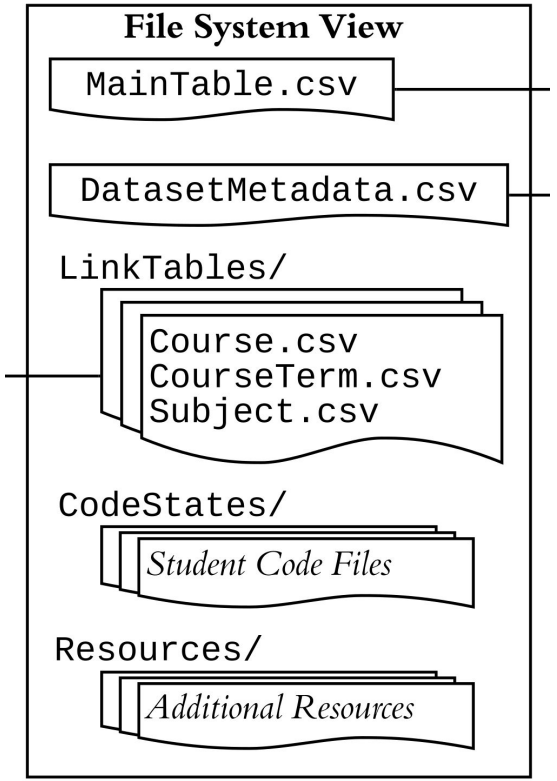
Data Format: Structure and Semantics



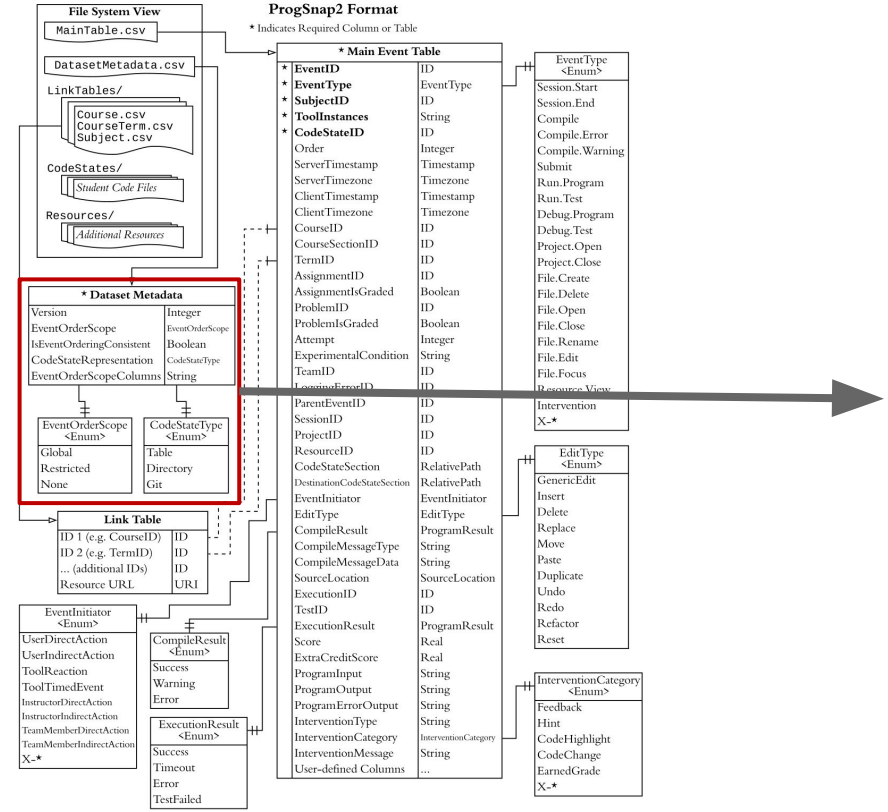
- File layout
- Dataset metadata
- Main event table
- Link tables

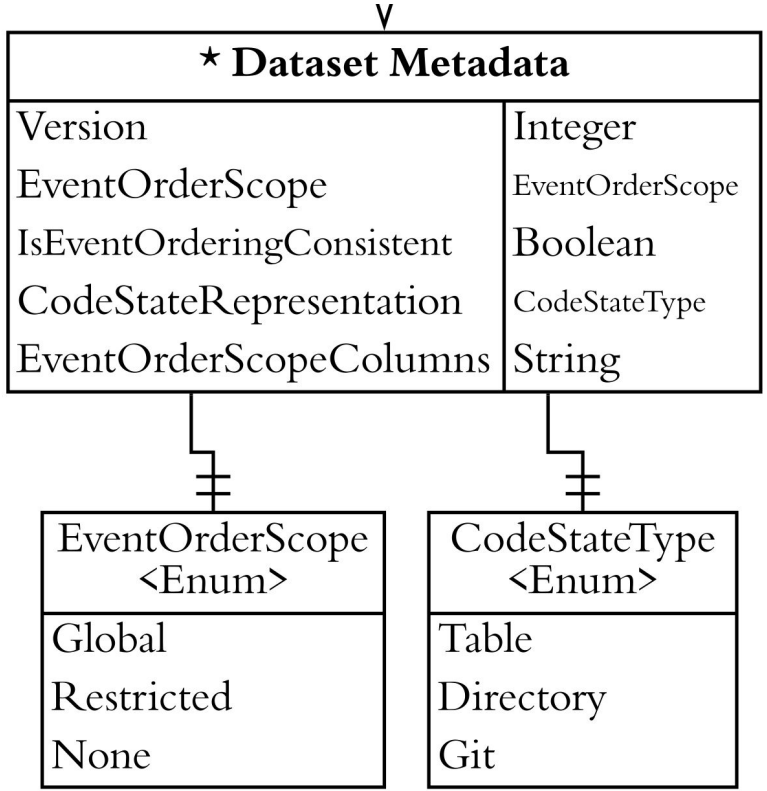
Data Format: File System



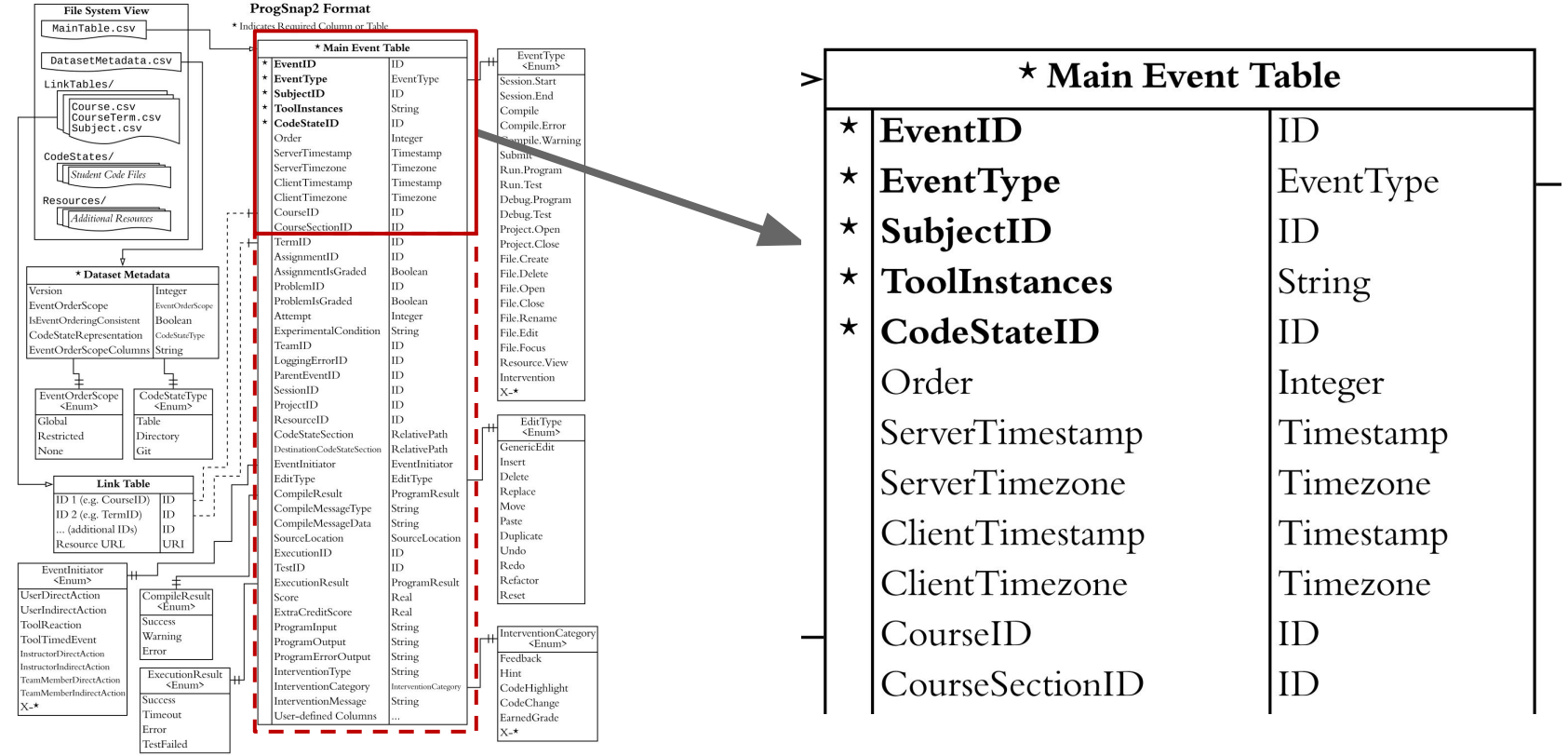


Data Format: Metadata





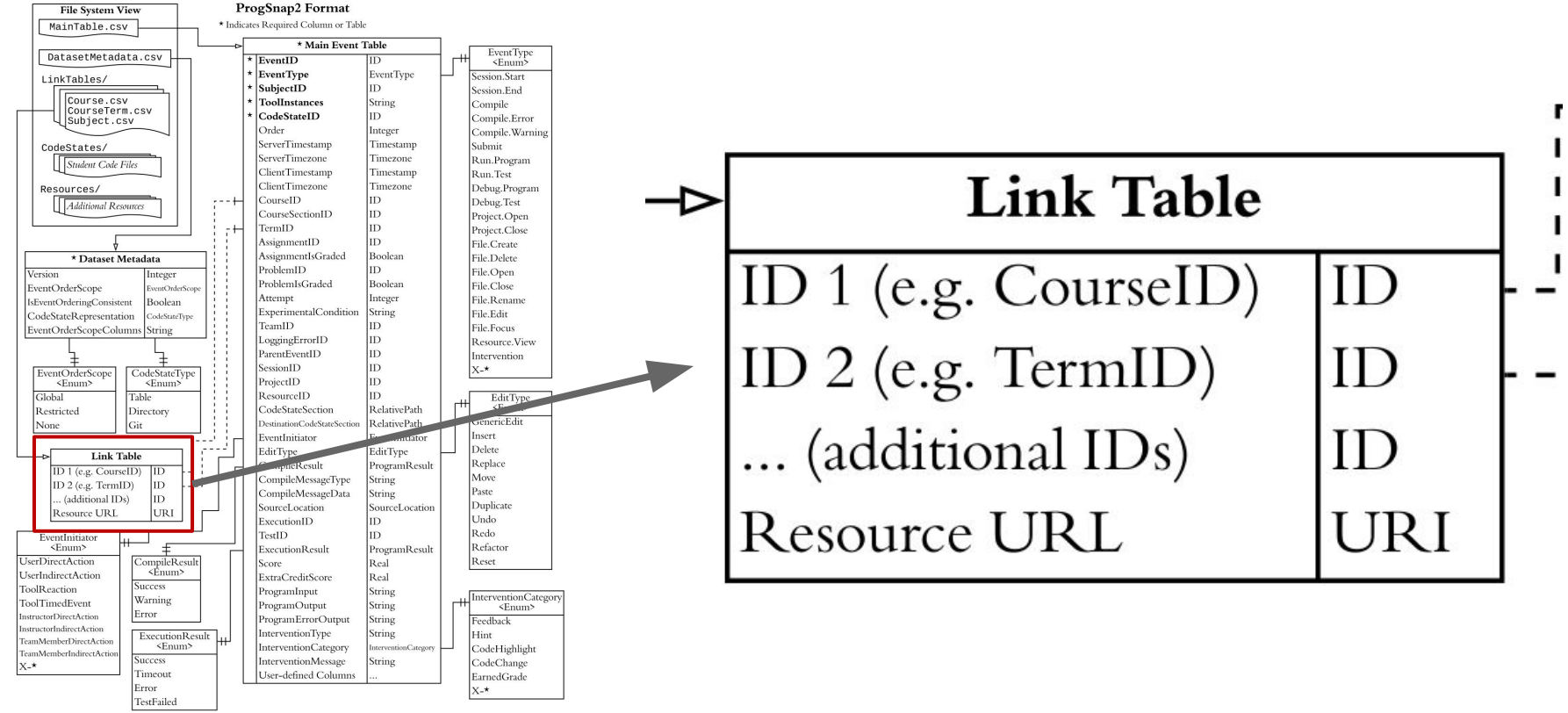
Data Format: Main Event Table



* Main Event Table

★	**EventID**	ID
★	**EventType**	EventType
★	**SubjectID**	ID
★	**ToolInstances**	String
★	**CodeStateID**	ID
	Order	Integer
	ServerTimestamp	Timestamp
	ServerTimezone	Timezone
	ClientTimestamp	Timestamp
	ClientTimezone	Timezone
	CourseID	ID
	CourseSectionID	ID

Data Format: Link Tables



Case Study: Compiler Error Metrics

Goal: Evaluate how well ProgSnap2 facilitates answering authentic research questions with data

Compiler Error Metric: Quantifies a student's struggle with compiler errors based on log data

- **EQ**: Error Quotient (Jadud, 2006)
- **Watwin** Score (Watson et al., 2013)
- **RED**: Repeated Error Density (Becker, 2016)

T.W. Price et al. “ProgSnap2: A Flexible Format for Programming Process Data.” Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education.

Datasets

We used 5 datasets from CS0/CS1 courses:

- Collected from **different systems and universities**
  - Varied in programming language, size, students, etc.
- 3 could not be shared with the analysis authors

System	CC	CWO	BlockPy	PCRS	ITAP
Language	C	Java	Python	Python	Python
Students	90 (-4)	410 (-3)	647 (-6)	1192 (-56)	73 (-16)
Exercises	86	50	244	99	38

Results: Correlations

All error metrics predicted students' grades:

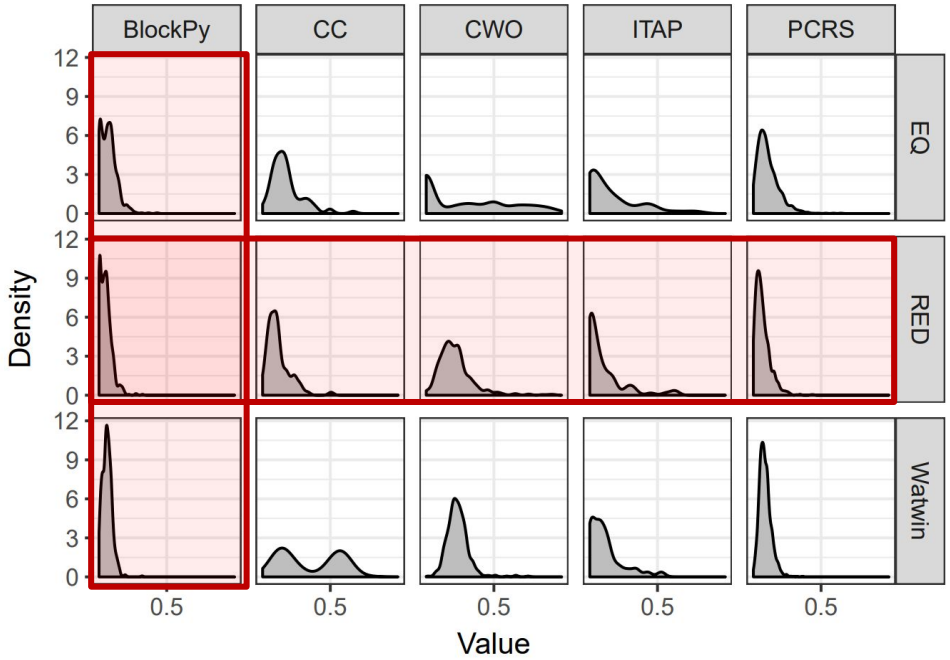
- Varied across datasets
- Weak correlations, not highly predictive

All error metrics are highly correlated.

Dataset		EQ	RED	Watwin
CC	RED	0.988***		
	Watwin	0.963***	0.988***	
	Grade	-0.409***	-0.467***	-0.374**
CWO	RED	0.975***		
	Watwin	0.871***	0.903***	
	Grade	-0.363***	-0.357***	-0.300***
BlockPy	RED	0.991***		
	Watwin	0.860***	0.854***	
	Grade	-0.254***	-0.244***	-0.190***
PCRS	RED	0.983**		
	Watwin	0.923***	0.912***	
ITAP	RED	0.946***		
	Watwin	0.900***	0.788***	

Significance codes (p <): * = 0.05; ** = 0.01; *** = 0.001

Results: Distributions



Distribution of metrics varied across **datasets**, but not across **metrics**.

Discussion

ProgSnap2 simplified analysis:

- A single script let us consistently apply the metrics
- Analysis code ran smoothly on unseen datasets
- Converting to the format took minimal effort

It did not remove some challenges of research:

- Still had to clean data, verify results, fix bugs

Future Work

ProgSnap2 has the potential to greatly facilitate:

- Collaboration
- Data sharing
- Gaining insight

The next step is **wider adoption**. We're looking for collaborators to:

- Produce / convert data sets to ProgSnap2 format
- Develop new logging systems guided by ProgSnap2
- Use and contribute to cross-dataset analysis code

SPLICE can help fund this work through mini grants!

How to Get Involved

ProgSnap2 was developed by a working group from CS-SPLICE (cssplice.org)

- Join the discussion: email David Hovemeyer (daveho@cs.jhu.edu)
- Check out the spec: bit.ly/ProgSnap2
- Code & public datasets: github.com/thomaswp/ProgSnap2Analysis