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 Gusukukuma, 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.

```cpp
#include <iostream>

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

*print("Hello, World!")*
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
ProgSnap2 Format

Data Format: Structure and Semantics

- File layout
- Dataset metadata
- Main event table
- Link tables
Data Format: File System
Data Format: Metadata

**Version**
Integer

**EventOrderScope**
EventOrderScope

**IsEventOrderingConsistent**
Boolean

**CodeStateRepresentation**
CodeStateType

**EventOrderScopeColumns**
String

---

**Progsnap2 Format**

**Data Format: Metadata**

* Dataset Metadata

- **Version**
- **EventOrderScope**
- **IsEventOrderingConsistent**
- **CodeStateRepresentation**
- **EventOrderScopeColumns**

---

**Progsnap2 Format**

**Main Event Table**

- **EventID**
- **EventScope**
- **SubjectID**
- **ToolInstance**
- **CodeStateID**
- **Order**
- **ServerTimestamp**
- **ClientTimestamp**
- **SessionID**
- **CourseSectionID**
- **TendID**
- **Assignments**
- **AssignmentGraded**
- **ProblemID**
- **ProblemGraded**
- **Mode**
- **ExperimentalCondition**
- **TeamID**

---

**Related Resources**

- **File System View**
- **Database Metadata**
- **Link Table**
- **Event Order Scope**
- **Code State Representation**

---

**Program2: Format for Programming Data**

Price et al.

ITiCSE 2020
Data Format: Main Event Table
Data Format: Link Tables

### ProgSnap2 Format

#### Event Table
- **EventID**: ID
- **EventType**: ID
- **SubjectID**: ID
- **ToolInstanceID**: String
- **CodeShareID**: ID
- **Order**: Integer
- **OccurrenceTime**: Timestamp
- **ServerTimezone**: String
- **ServerTimezoneOffset**: String
- **ClientTimezone**: String
- **EventName**: String
- **ContractID**: ID
- **OccurredSectionID**: ID
- **OccurredTime**: String
- **OccurredLocation**: String

#### Assignment Table
- **AssignmentID**: ID
- **AssignmentGraded**: Boolean
- **ProblemID**: ID
- **ProblemGraded**: Boolean
- **Message**: String
- **AssignmentDueDate**: Date
- **AssignmentDeadline**: Date
- **AssignmentDueTime**: Time
- **AssignmentDeadlineTime**: String

#### Resource Table
- **ResourceId**: ID
- **ResourcePath**: String
- **ResourceName**: String
- **ResourceId**: ID
- **ResourceSection**: RelationalPath
- **ResourceSection**: String
- **ResourceType**: String
- **ResourceUri**: URI

#### Link Table
- **ID 1 (e.g. CourseID)**
- **ID 2 (e.g. TermID)**
- ... (additional IDs)
- **Resource URL**
- **URI**
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)

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

<table>
<thead>
<tr>
<th>System</th>
<th>CC</th>
<th>CWO</th>
<th>BlockPy</th>
<th>PCRS</th>
<th>ITAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>C</td>
<td>Java</td>
<td>Python</td>
<td>Python</td>
<td>Python</td>
</tr>
<tr>
<td>Students</td>
<td>90 (-4)</td>
<td>410 (-3)</td>
<td>647 (-6)</td>
<td>1192 (-56)</td>
<td>73 (-16)</td>
</tr>
<tr>
<td>Exercises</td>
<td>86</td>
<td>50</td>
<td>244</td>
<td>99</td>
<td>38</td>
</tr>
</tbody>
</table>
Results: Correlations

All error metrics predicted students' grades:
- Varied across datasets
- Weak correlations, not highly predictive

All error metrics are highly correlated.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>EQ</th>
<th>RED</th>
<th>Watwin</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>RED</td>
<td>0.988***</td>
<td>0.988***</td>
</tr>
<tr>
<td></td>
<td>Watwin</td>
<td>0.963***</td>
<td>0.988***</td>
</tr>
<tr>
<td>Grade</td>
<td>-0.409***</td>
<td>-0.467***</td>
<td>-0.374***</td>
</tr>
<tr>
<td>CWO</td>
<td>RED</td>
<td>0.975***</td>
<td>0.903***</td>
</tr>
<tr>
<td></td>
<td>Watwin</td>
<td>0.871***</td>
<td>0.903***</td>
</tr>
<tr>
<td>Grade</td>
<td>-0.363***</td>
<td>-0.357***</td>
<td>-0.300***</td>
</tr>
<tr>
<td>BlockPy</td>
<td>RED</td>
<td>0.991***</td>
<td>0.854***</td>
</tr>
<tr>
<td></td>
<td>Watwin</td>
<td>0.860***</td>
<td>0.854***</td>
</tr>
<tr>
<td>Grade</td>
<td>-0.254***</td>
<td>-0.244***</td>
<td>-0.190***</td>
</tr>
<tr>
<td>PCRS</td>
<td>RED</td>
<td>0.983**</td>
<td>0.912***</td>
</tr>
<tr>
<td></td>
<td>Watwin</td>
<td>0.923***</td>
<td>0.912***</td>
</tr>
<tr>
<td>ITAP</td>
<td>RED</td>
<td>0.946***</td>
<td>0.788***</td>
</tr>
<tr>
<td></td>
<td>Watwin</td>
<td>0.900***</td>
<td>0.788***</td>
</tr>
</tbody>
</table>

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 (csssplice.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