




## Visually Explaining Source Code in CS Education

**Benjamin Biegel, Sebastian Baltes, Bob Prevos, and Stephan Diehl**

**University of Trier, Germany**

 @s\_baltes

 s.baltes@uni-trier.de

IEEE Symposium on  
Visual Languages and Human-Centric Computing  
October, 18-22 – Atlanta, USA



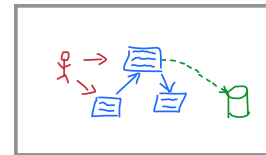
# Motivation

- In CS education, teachers have to present concrete source code examples and related abstract concepts
- Traditional presentation tools have shortcomings:



# PowerPoint

- Easy-to-use drawing features (shapes, colors, etc.)
- But: Linear, predefined presentation
- Interaction with audience?
- Source code formatting?



# Whiteboards

- Flexible, but depending on drawing skills of teacher
- Difficult to prepare or modify content
- Not suitable for longer source code examples

```
2 #?php
3 echo "<?xml version='1.0' encoding='iso-8859-2'?>";
4 // double quote string - >> //
5 let $obj = new stdClass();
6 $obj->double_quote_string = "[someobject->result]=[a]";
7 [someobject->result]=[a] [t1] 813 error in
8 8 10 18 18
9 $t = 2.1 // error
10 $t = 2.1 + .2 // numbers
11 }
12 $output = <<< what_ever_you_TYPE
13 what_ever_you_TYPE ;
14 what_ever_you_TYPE;
15 what_ever_you_TYPE;
16 what_ever_you_TYPE;
17
18 while ( my_function($arg) 64 mysql_query($query) ) { //
19 }
20 }
```

## Code Editors / IDEs

- Exploration and modification of source code possible
- But: Not designed for presenting source code (e.g. step-by-step revealing)
- Visualization difficult



# Motivation

## Tradeoff:

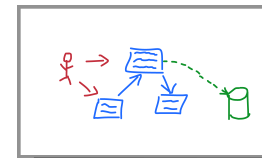
visually appealing but static



flexible but text-oriented

```
2 <?php
3 echo "<?xml version='1.0' encoding='iso-8859-2'>";
4 /** doc-style comment - ?> */
5 if( SOME_CONSTANT ) {
6     fo = "double quote string; fo=object->result->a[0]
7     fo=object->result->a[0]; fo[2]; $![] error in
8     fo = '10 88';
9     fo = 2.1; // error
10    fo = 2.1 + .2; // numbers
11 }
12 $output = <<< what_ever_you_TYPE
13 what_ever_you_TYPE;
14 what_ever_you_TYPE;
15 what_ever_you_TYPE;
16 what_ever_you_TYPE;
17
18 while( my_function($arg) && mysql_query($query) ){ //
19 }
20 }
```

flexible but cumbersome



## Common scenario:

Switching between PowerPoint and source code editor



**Our goal:** New tool for presenting source code combining:

- Drawing features of PowerPoint
- Flexibility of source code editors
- Interactivity of whiteboards

# Our Approach

---

# Requirements

## Preparation



- Create content in advance
- Prepared content can be reused

## Presentation



- Step-by-step revealing of visuals and code
- Highlighting during lecture

## Follow-up



- Content available to students immediately after lecture

## Flexibility



- Modifying and adding content during the lecture

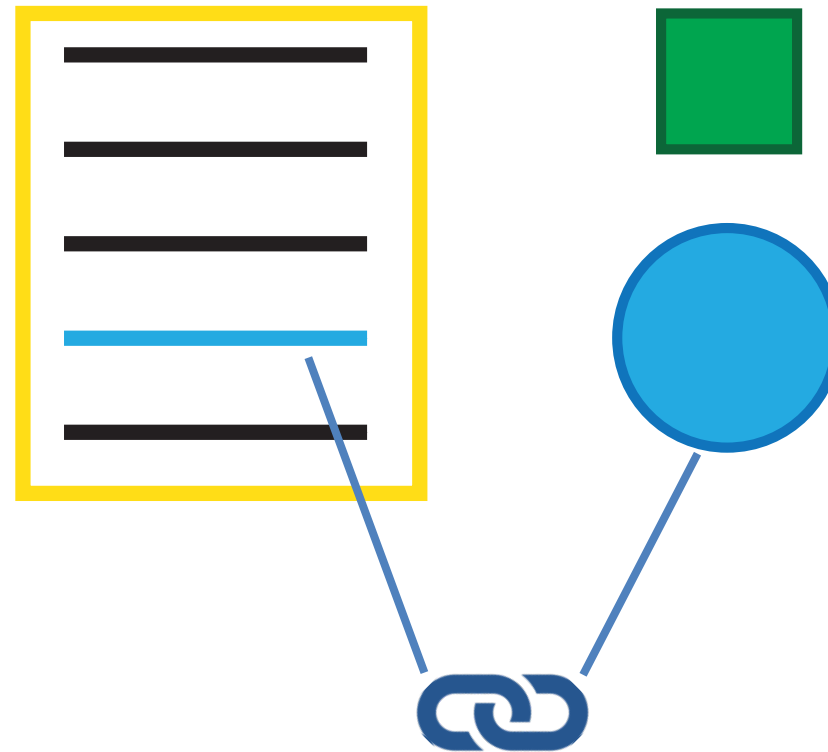
- Digital canvas with source code editor
- Teacher defines beforehand in which order source code is revealed
- Drawing features similar to PowerPoint, but intended to be used also during lecture
- Drawings and code can be linked  
→ Position is updated if code is inserted
- Highlighting features for both source code and drawings
- Presentation can be saved and shared any time during the lecture
- Students can branch documents for own annotations



- Digital canvas with source code editor
- Teacher defines beforehand in which order source code is revealed
- Drawing features similar to PowerPoint, but intended to be used also during lecture
- **Drawings and code can be linked**  
→ **Position is updated if code is inserted**
- Highlighting features for both source code and drawings
- Presentation can be saved and shared any time during the lecture
- Students can branch documents for own annotations

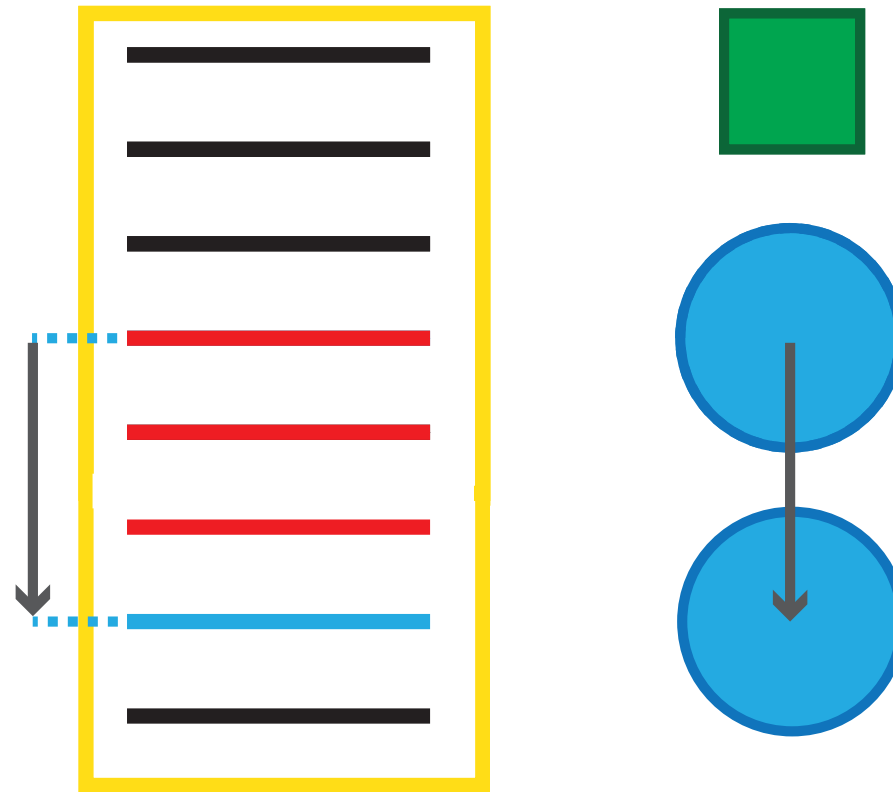


# Linking of Code and Drawings

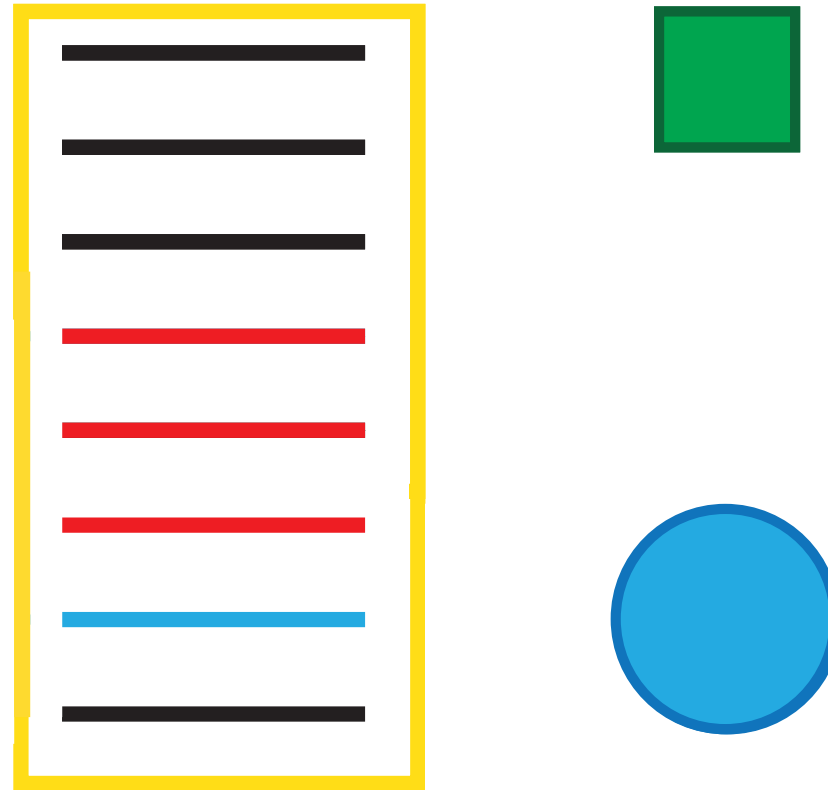




# Linking of Code and Drawings



# Linking of Code and Drawings



# Prototype Implementation

---

# Prototype Implementation

- Web-based
- Runs on both desktop and tablet browsers (Firefox and Safari)
- Source code “animation” using XML file

The screenshot displays a web-based IDE interface. On the left, a code editor shows the following Java code:

```
0 interface MenuVisitor {  
1     abstract void visit(MenuEntry entry);  
2     abstract void visit(MenuEntry2 entry);  
3 }  
4 class MenuEntry {  
5     public void accept(MenuVisitor visitor)  
6     { visitor.visit(this); }  
7 }  
8 class MenuEntry2 {  
9     MenuEntry2 entries[];  
10    public void accept(MenuVisitor visitor) {  
11        visitor.visit(this);  
12        for(int i=0;i<entries.length;i++)  
13        { entries[i].accept(visitor); }  
14    }  
15 }
```

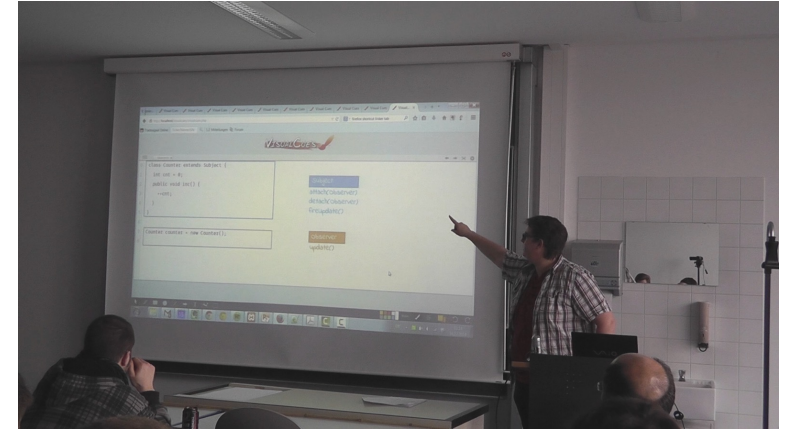
On the right, a diagram illustrates the Visitor pattern. A blue circle labeled "Menu" and an orange circle labeled "Visitor" are connected by two arrows: a grey arrow labeled "accept" pointing from Menu to Visitor, and a grey arrow labeled "visit" pointing from Visitor to Menu. Above the diagram is a toolbar with icons for anchoring, zooming, and other navigation functions. The interface includes a browser window titled "Visitor" and a toolbar at the bottom with various editing tools.

# Practical Experiences

---

## Evaluation in two undergraduate lectures:

1. Software engineering (design patterns)
  - Interviewed three students and two teachers
2. Programming concepts (binary search tree implementation)
  - Interviewed three students and two teachers
  - Questionnaire



We used an ad-hoc visual notation, often utilizing color to indicate relation:

# Practical Experiences

Observer

```
0 interface Observer {  
1  
2 }  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15
```

Visual Cues

Datenstruktur: Binärer Suchbaum

```
0  
1 class BNode {  
2     BNode parent;  
3     int key;  
4     BNode left;  
5     BNode right;  
6     BNode(int k) { key=k; }  
7 }  
8  
9 class BTree {  
10     BNode root;  
11     ... insert, minimum, successor, delete  
12 }  
13  
14  
15
```

cl

4?

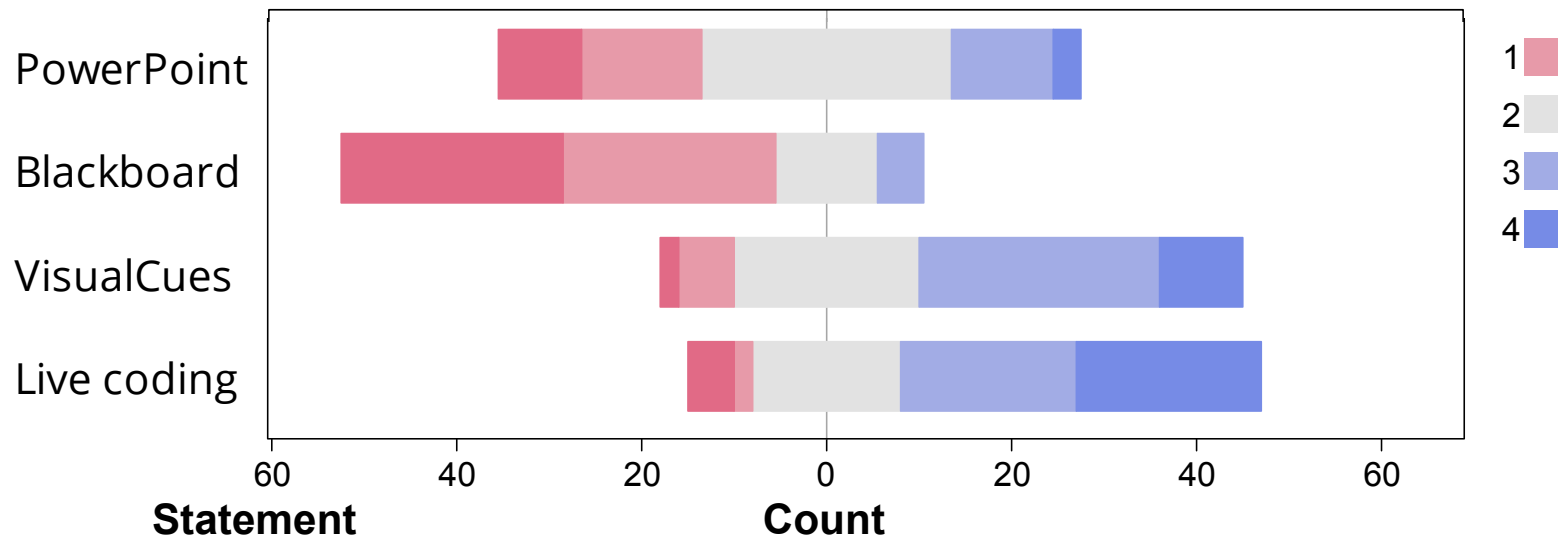
4

# Students' View

- Ad-hoc notation understandable
- Appreciated step-by-step revealing of source code and linking
- Liked that teacher can immediately respond to questions
- *"VisualCues prevents the teacher from doing PowerPoint karaoke."*
- *Questionnaire:*



"For future lectures, I would like source code examples to be presented mainly using..."



n=63

Likert scale items ranging from 0="I do not agree" to 4="I agree"



# Teachers' View

- 50% of drawings prepared, 50% created ad-hoc during lecture
- Often framed code with colored rectangle and used same color for corresponding drawings
- Tried to use shapes consistently
- Better support for planning the lecture requested




# Conclusion and Future Work

- Students not happy with status quo (presenting source code with PowerPoint)
- VisualCues was well received
- But also strong preference for live coding
- **Future work:**
  - Combine VisualCues and live coding
  - Evaluation in larger context
  - Make VisualCues also available for students during lecture?



Demo video and supplementary material:

<http://st.uni-trier.de/visualcues>

 @s\_baltes

 s.baltes@uni-trier.de

