Principled Assessment of Student Learning in High School Computer Science

Eric Snow, Daisy Rutstein, Marie Bienkowski, Yuning Xu
Center for Technology in Learning
SRI International

ACM International Computing Education Research (ICER) Conference
August 2017

Principled Assessment of Computational Thinking

NSF

SRI Education

Overview

- Significance & Need
- Background & Related Research
- Methodology
- Findings
- Discussion
- Conclusions

Significance & Need

- Computer science is spreading throughout the US K-12 system, with full, yearlong classes being offered in secondary schools
- Increased demand for assessments that support valid inferences about student learning

Significance & Need

* **Teachers in the introductory CS courses** >>> how should I adapt instruction to meet my students' needs?
* **Teachers in advanced CS courses** >>> how well are students prepared for advanced work and where do they need extra help?
* **Principals** >>> is my school offering rigorous CS courses?
* **Stakeholders** >>> what CS knowledge and skills students are developing?

Significance & Need

Development of high-quality assessments has not kept pace with the spread of CS programs/curricula throughout the US K-12 system

2015 study by Computer Science Teacher's Association (CSTA) found that teachers face a number of challenges finding valid and reliable assessments to use in their classrooms

Assessment Challenges

Challenge #1: Understanding the Domain

>>> What is important for computer scientists to know and be able to do? What are the important **practices** of CS?

Challenge #2: Developing Authentic Representations

>>> How can we develop tasks/situations that elicit evidence of computational thinking practices?

Challenge #3: Eliciting Valid Evidence

>>> Does the evidence support the inferences we want to make about computational thinking practices?

Challenge #1: Understanding the Domain

What are computational thinking practices?

Background & Related Research – Computational Thinking Practices

In the US, the new **K12 CS Framework and aligned CSTA standards,** and the **Common Core State Standards** and **Next Generation Science Standards** all include guidance related to **computational thinking practices.**

Background & Related Research – Computational Thinking Practices

This reflects an orientation toward not just an internal, individual "thinking" but **"ways of being and doing" that students should *demonstrate*** when learning and exhibiting computer science knowledge, skills, and attitudes.

It represents the ***application*** of CS content knowledge via problem solving and inquiry-based methods.

Challenge #2: Developing Authentic Representations

How can we develop tasks/situations that elicit evidence of computational thinking practices?

Background & Related Research – Evidence-Centered Design (ECD)

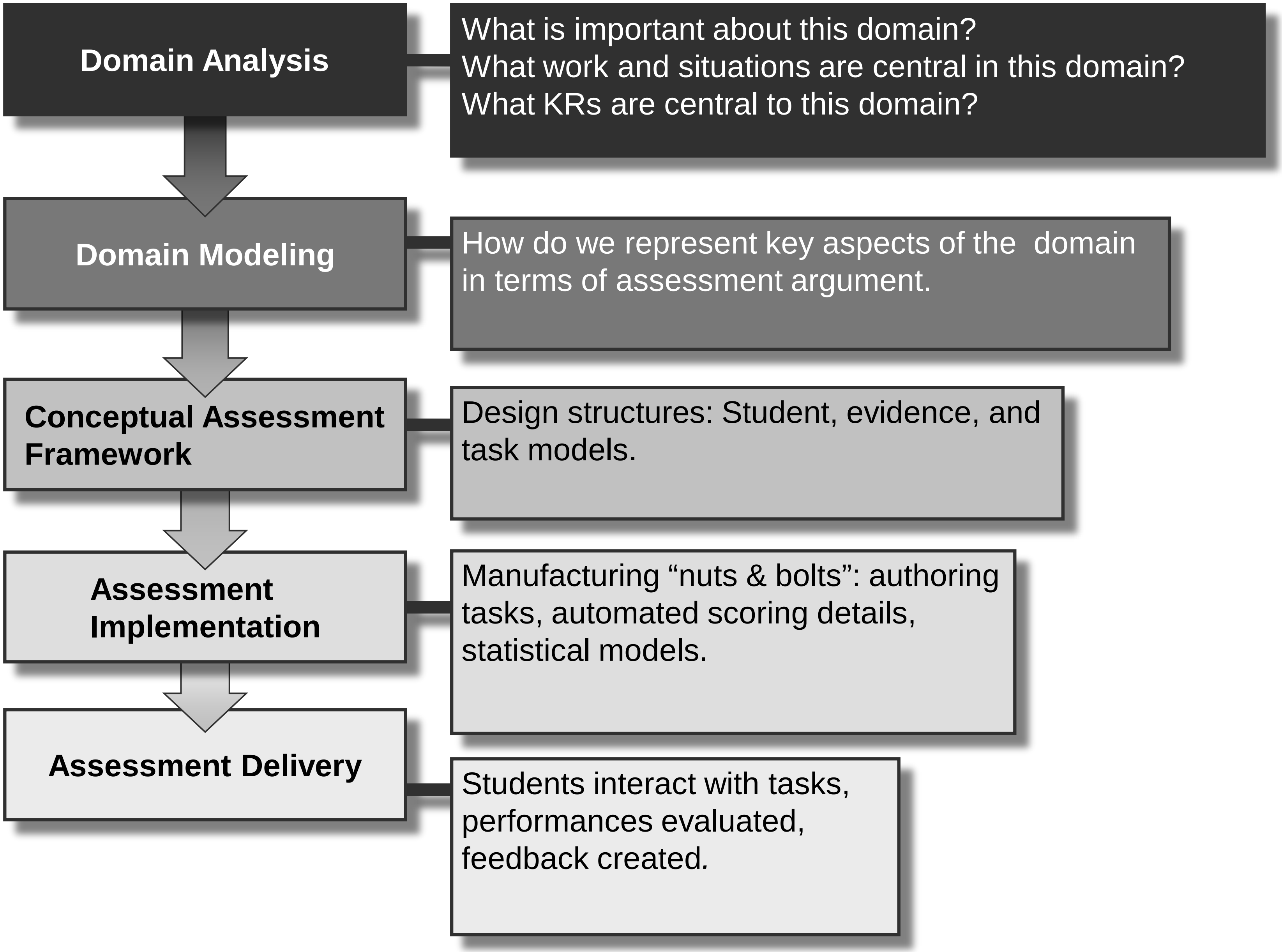
ECD is a framework for assessment design and development

Used by major testing companies and is a requirement of many states for developing assessments

Views assessment as a process of gathering evidence to support inferences about what a student knows and can do

Provides a structure for an approach that incorporates validity evidence into the assessment design process

Particularly useful when the knowledge/skills to be measured involve complex, multistep performances, such as those required in computational thinking practices



- From Mislevy & Riconscente, 2006

Layers of Evidence-Centered Design and Key Entities for Computational Thinking

ECD Layer	Activity	Key Entities & Examples
Domain analysis	Gather substantive information about the domain of interest that has implications for assessment; how knowledge is constructed, acquired, used, and communicated	Computational thinking domain concepts (e.g., abstraction, automation); terminology (debugging); tools (programming languages); representations (storyboards); situations of use (modeling predator-prey, visual storytelling), and curriculum standards and mappings
Domain modeling	Express assessment argument in narrative form based on information from domain analysis	Specification of knowledge, skills, and other attributes to be assessed (e.g., describe result of running a program on given data); features of situations that can evoke evidence (find errors in programs); kinds of performances that convey evidence (use of recursion)

- From Mislevy & Riconscente, 2006

Layers of Evidence-Centered Design and Key Entities for Computational Thinking, cont’d

ECD Layer	Activity	Key Entities & Examples
Conceptual assessment framework	Express assessment argument in structures and specifications for tasks and tests, evaluation procedures, measurement models	Student, evidence, and task models; student, observable, and task variables; rubrics; measurement models; test assembly specifications; task templates and task specifications
Assessment implementation	Implement assessment, including presentation-ready tasks and calibrated measurement models	Tasks, task materials (including supporting materials, tools, affordances); pilot test data to hone evaluation procedures and fit measurement models
Assessment delivery	Coordinate interactions of students and tasks: task- and test-level scoring; reporting	Tasks as presented; work products as created; scores as evaluated

- From Mislevy & Riconscente, 2006

Background & Related Research – Evidence-Centered Design

More information about ECD can be reviewed at:

https://ecd.sri.com/

Challenge #3: Eliciting Valid Evidence

Does the evidence support the inferences we want to make about computational thinking practices?

Background & Related Research – Test Validity

The latest thinking in test validity focuses on supporting assessment inferences through collecting and integrating different types of evidence:

* Test Content
* Internal Structure
* Response Processes
* Relations to other Variables
* Test Use

Principled Assessment of Computational Thinking

How can we improve CS teaching, learning, and adoption through evidence-centered assessment?

Context – *Exploring Computer Science* (ECS)

* Pre-AP, introductory CS course
* Late middle school / early high school
* Six, six-week units
* Focus on **equity**
* A central tenet of ECS pedagogy is **inquiry-based learning**: core concepts learned through induction, teaching through guided inquiry, and open-ended assessments

Methodology –
Analyzing and Modeling the CTP Domain for ECS

Domain Analysis

What is important about the CTP domain for ECS?
What work and situations are central in CTP domain for ECS?
What KRs are central to the CTP domain for ECS?

* *The CTP domain as it is represented in ECS.*

Methodology –
Analyzing the CTP Domain for ECS

Standards/Curriculum

- Exploring Computer Science, Unit 1-4 Learning Objectives
- CSTA (2011). CSTA K-12 Computer Science Standards
- College Board (2010). AP CS Principles: Big Ideas, Key Concepts, and Supporting Concepts
- NGSS, CCSS

Literature

- National Academies Report: Computer Science: Reflections on the Field, Reflections from the Field
- SIGCSE, CSTA, ITiCSE, Journal of Computing in Higher Education, Educational Researcher
- Jeanette Wing & others; National Academies Workshop on Pedagogical Aspects of Computational Thinking

ECS Units	Computational Thinking Practices
Unit 1: Human-Computer Interaction	• Analyze the effects of developments in computing.
Unit 2: Problem Solving	• Design and implement creative solutions and artifacts. • Apply abstractions and models. • Analyze their computational work and the work of others.
Unit 3: Web Design	• Design and implement creative solutions and artifacts. • Analyze their computational work and the work of others. • Connect computation with other disciplines.
Unit 4: Introduction to Programming	• Design and implement creative solutions and artifacts. • Apply abstractions and models. • Analyze their computational work and the work of others.

Methodology –
Modeling the CTP Domain for ECS

Domain Modeling

How do we represent key aspects of the CTP domain for ECS in terms of an assessment argument?

Methodology –
Modeling the CTP Domain for ECS

Developed assessment **design patterns** for ECS units 1-4:

- Human-computer interaction
- Problem solving
- Web design
- Introduction to programming

Design patterns are high level representations of an assessment argument that **lay out a design space** for assessment developers

ECS Unit / Computational Thinking Practice	Example ECS Unit Focal Knowledge, Skills, Attributes (FKSAs)	Example CTP Focal Knowledge, Skills, Attributes (FKSAs)
Unit 1: Human-Computer Interaction ↕ Analyze the effects of developments in computing.	• Students are able to explain why an object is or is not a computer.	• Ability to describe the characteristics of a computer (including what it means for a computer to be "intelligent").
	• Students are able to evaluate the implications of a form of data exchange on social interactions.	• Ability to analyze the effects of computing on society within economic, social, and cultural contexts.
	• Students are able to explain how computing innovation has led to new types of legal and ethical concerns.	• Ability to evaluate legal and ethical concerns raised by computing-enabled innovations.

Methodology –
Modeling the CTP Domain for ECS

Example FKSA	Example Potential Work Product (what students produce)	Example Potential Observations (evidence in what students produce)
Students are able to explain why an object is or is not a computer.	An explanation of why an object is or is not a computer.	Appropriateness of the explanation of why an object is or is not a computer. • Did the student correctly identify aspects of the object that relate to aspects of a computer? • Did the student correctly identify aspects of a computer that the object lacks?

Methodology –
Modeling the CTP Domain for ECS

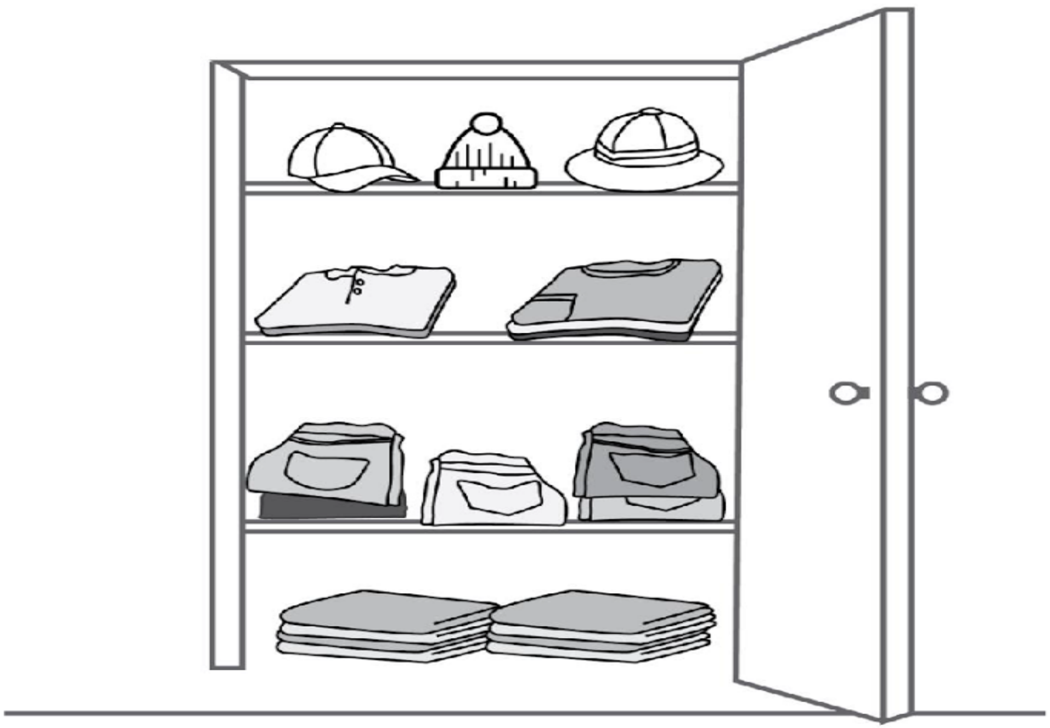
Example FKSA	Example Characteristic Features	Example Variable Features
Students are able to explain why an object is or is not a computer.	The student must be presented with an object The object must have clear characteristics that allow the evaluation of whether it is a computer.	Whether the object could be considered a computer or not. Whether students would be able to argue either way if the object is a computer or not. The degree to which the important characteristics are explicitly stated in the problem or must be inferred by the test taker.

Methodology – Developing Assessments of CTPs for ECS

* Extended, scenario-based tasks containing 3-4 embedded forced-choice and short constructed response items
* Present students with more authentic situations to measure CTPs
* Focus on applying, analyzing, evaluating levels of Bloom's taxonomy.
* Other Bloom levels (e.g., create) and corresponding FKSAs were measured using other formats (e.g., portfolios)

2. Carla programmed a robot to select clothes for her. The robot is able to move around the room, open and close doors, and pick up and drop objects.

Carla's Closet



Below is a set of instructions for the robot once the robot is inside the closet:

1. Take out the top pair of pants from the left side of the third shelf down.
2. Take out a T-shirt.
3. Take out the hat from the top shelf that matches the outfit.

Sample Responses

2-Point Sample Response 1

For sample response 1,

a) Student selects "Take out a t-shirt" and correctly explains that the instruction does not say exactly what type of t-shirt.

b) Student correctly re-writes step to include location of where the shirt to select is in the closet.

a)	**Step:** Take out a t-shirt A robot would have difficulty completing this step because the instructions don't say what exact type of shirt. You can have many types of shirts in the closet. Shirts come in different colors, styles, and sizes.
b)	Take out first t-shirt from the right hanging on hanger.

1-Point Sample Response 2

For sample response 2,

a) 1pt: Student "Take out the hat from the top shelf that matches the outfit" and correctly explains a robot does not have fashion sense or knows how to determine to match a hat with an outfit.

b) 0pts: Student incorrectly re-writes step which specifies clothing item, but does not give precise/complete instructions about hat selection.

a)	**Step:** Take out the hat from the top shelf that matches the outfit. A robot wouldn't have a fashion sense, or know how to determine which hat would match, it wouldn't have a mind of its own.
b)	Take out the hat that matches the red shirt.

0-Point Sample Response 4

For sample response 4,

a) Student "Take out a t-shirt" and provides an incomplete explanation.

b) Student incorrectly re-writes step with vague instructions.

a)	**Step:** Take out a t-shirt. How would a robot know what shirt to pick? Maybe the classmate or the person who made the robot thinks a different shirt is better. Or the shirt might not match the outfit.
b)	Select the most appropriate shirt for my outfit.

Methodology – Piloting & Refining Assessments for ECS

* Pilot 1 2014-2015, Pilot 2 2015-2016
* ECS teachers from across the U.S. including Los Angeles, Chicago, and New York
* Collected validity evidence based on test content and student responses processes to help us refine and improve the assessments
  * Test content >>> expert review of alignment between the knowledge and skills, the curriculum learning goals, and CT practices
  * Student response processes >>> cognitive think-aloud interviews with students participating in the pilot testing activities

Methodology – Scoring & Inter-Rater Reliability

- Researchers were trained on the rubrics
- Each assessment was scored by two different scorers with a third scorer scoring if there were discrepancies in the scores
- Inter-rater reliability was high, with over 90% agreement between raters for most of the tasks
- Tasks for which the reliability was lower were revised either by modifying the item to clarify what was expected or by modifying the rubric

Analysis & Results –
Descriptive Statistics

Table 1: Summary of Assessment Information

Assessment	Content	Form	# of Tasks	# of Common Tasks	# of Items	Total Points	# of Teachers	# of Students
Unit 1	Human-Computer Interaction	Form A	10	6	22	29.00	5	59
		Form B	10		22	29.00	5	53
Unit 2	Problem Solving	Form A	10	-	24	29.50	5	163
Unit 3	Web Design	Form A	8	5	19	27.00	12	189
		Form B	8		20	31.00	12	208
Unit 4	Introduction to Programming	Form A	9	7	21	33.00	13	192
		Form B	8		19	30.00	13	196
Cumulative	Units 1-4	Form A	6	2	17	21.00	7	139
		Form B	7		19	22.00	7	142

Analysis & Results – Descriptive Statistics

* ~ 40% female/ 60% males, ~50% Hispanic/Latino (49.28%)
* Average total scores in the 60 - 70% range across the assessments
* Female and male students had comparable average scores on the assessments
* Score distributions were slightly negatively skewed, indicating more students scored at the high end of the score distributions.

Analysis & Results –
Validity Evidence Based on Internal Structure

Table 2: Validity Evidence Based on Internal Structure

		# of Students	Cronbach's Alpha	Assessment Difficulty	Task Difficulty		Average Task Discrimination
					Min	Max	
Unit 1	Form A	59	.82	.67	.10	.86	.37
	Form B	53	.83	.58	.06	.85	.50
Unit 2	Form A	163	.84	.63	.45	.83	.55
Unit 3	Form A	189	.69	.67	.41	.83	.44
	Form B	208	.69	.64	.46	.82	.44
Unit 4	Form A	192	.49	.59	.58	.91	.44
	Form B	196	.76	.62	.31	.91	.46
Cumulative	Form A	139	.66	.65	.50	.74	.32
	Form B	142	.68	.64	.45	.77	.34

Analysis & Results –
Validity Evidence Based on Internal Structure

* Moderate to high levels of reliability (.66 - .84)
* Factor analysis supported expected structure of unit and cumulative assessments
* Moderate task difficulty levels, with the index ranging from .58 to .67
* High discriminating power for tasks/items with medium levels of difficulty

Discussion

Validity evidence based on internal structure is particularly promising:

* tasks within each unit assessment are all measuring one general construct
* assessments best suited for differentiating students of average ability

Discussion

Next Steps

* Examine whether validity results hold w/ larger sample and schools from different contexts
* Developing additional assessment tasks, particularly those with easy and hard levels of difficulty to improve utility across wider range of ability levels
* Item Response Theory (IRT) and Testlet Response Theory (TRT) analyses

Conclusions

Important effort to apply principled assessment design methods and contemporary test-validity standards to guide the development, piloting and validation of assessments of CTPs

Conclusions

Support use of the assessments by both educators measuring students' CT practices and by researchers studying curriculum implementation and student learning in introductory high school computer science

Conclusions

ECS teachers and other groups can use design patterns to develop novel assessment items for use in their classroom or in research on the impact of the ECS curriculum on students’ CT practices

New PACT Report:

Snow, E., Tate, C., Rutstein, D., Bienkowski, M. (2017). *Assessment design patterns for computational thinking practices in Exploring Computer Science.**

* To be released September 2017 on the PACT web site, http://pact.sri.com

Principled Assessment of Computational Thinking

NSF

SRI Education

More information about PACT?

https://pact.sri.com/

More information about the CTP Design Patterns?

http://bit.ly/2u6t0Nw

Review the ECS assessments and rubrics?*

https://pact.sri.com/ecs-assessments.html

* Terms of Use & Licensing Information: https://pact.sri.com/assessment/termslicense.html

Principled Assessment of Computational Thinking

NSF

SRI Education

THANK YOU!

This work was conducted with support from the National Science Foundation under contract numbers, CNS-1132232, CNS-1240625, and DRL-1418149.

Principled Assessment of Computational Thinking

NSF

SRI Education

Questions?

Principled Assessment of Computational Thinking

NSF

SRI Education