Principled Assessment of Computational Thinking

Principal Investigators: Eric Snow & Marie Bienkowski, SRI International. Partner: University of Oregon, Exploring Computer Science

Goals

- Develop a valid and reliable assessment of computational thinking
- Aid in the adoption of high school computer science courses through assessments that stakeholders recognize as useful boundary objects
- Create **design patterns** for major computational thinking concepts that can be used to develop new assessments as curriculum evolves
- Create and field test assessments for *Exploring Computer Science*

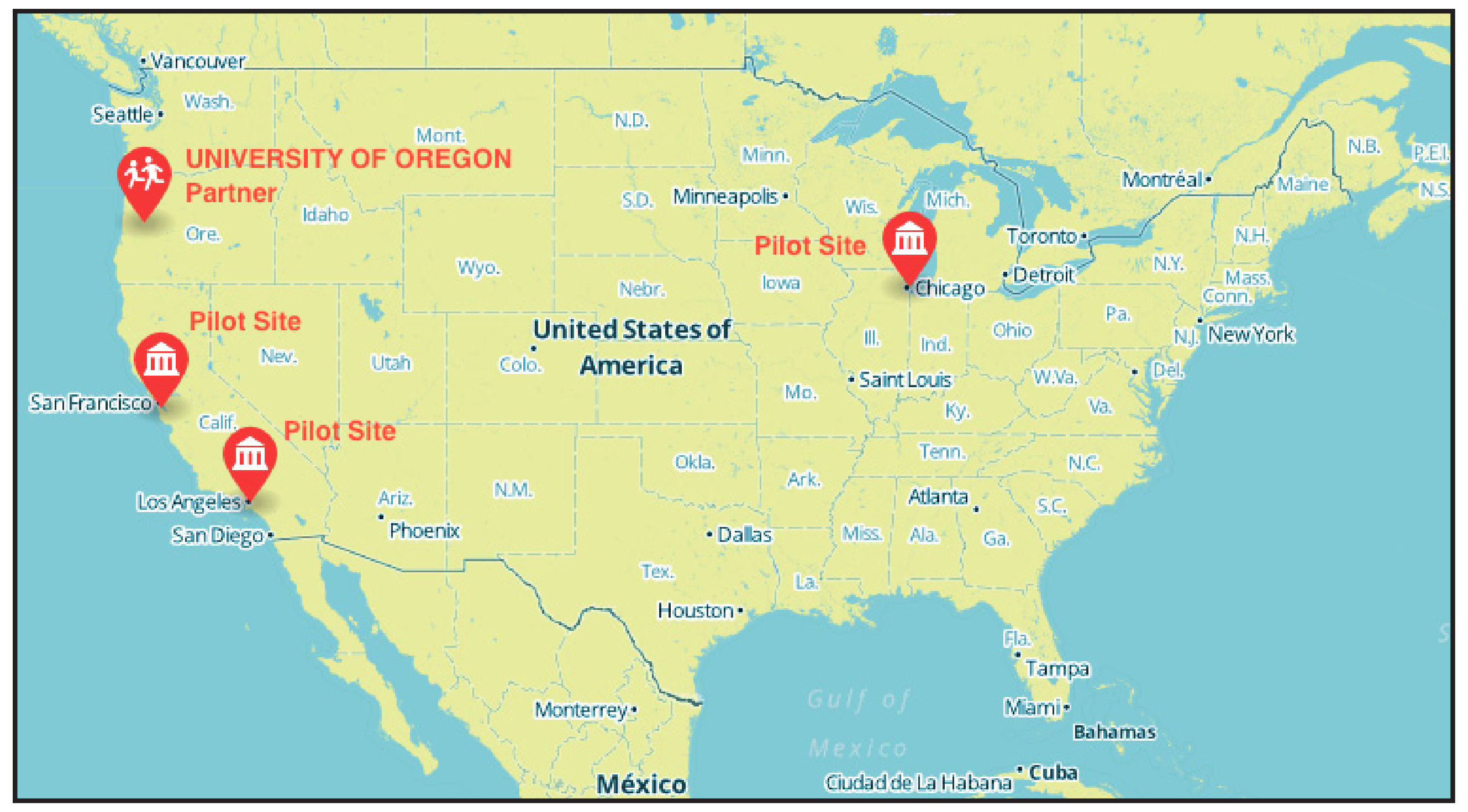
Activities

- Align *Exploring Computer Science* lesson objectives to CSTA, NETS, Common Core, NGSS, and state science and CTE standards
- Create conceptual assessment framework
- Define computational thinking practices (CTP) and focal knowledge, skills, and abilities (FKSAs) that constitute them
- Develop and apply CTP design patterns to guide the development of assessments for *Exploring Computer Science*
- Field test *Exploring Computer Science* assessments

Focal KSAs are the core of design templates which specify what to measure, the behaviors that are evidence of what we want to measure, and the types of tasks we can use to elicit the desired behaviors.

Scope

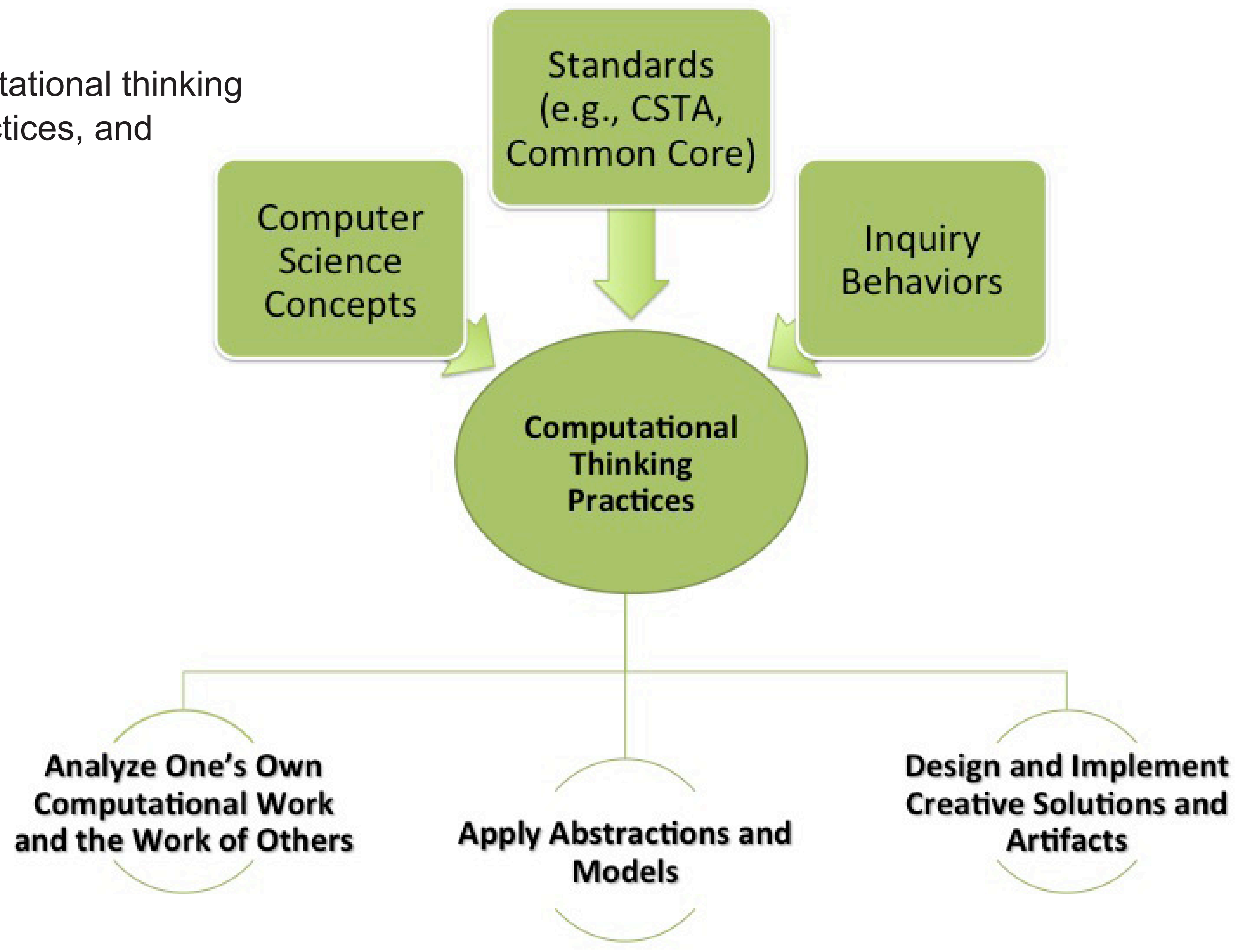
Partner sites for developing and field testing assessments for the *Exploring Computer Science* curriculum.



Creating boundary objects for the ECS curriculum: How can we improve CS teaching, learning, and adoption through evidence-based assessment?

Outcomes

More precise definitions of computational thinking practices, design patterns for practices, and establishing field test sites.


Standards (e.g., CSTA, Common Core)
Computer Science Concepts
Inquiry Behaviors
Computational Thinking Practices
Analyze One's Own Computational Work and the Work of Others
Apply Abstractions and Models
Design and Implement Creative Solutions and Artifacts

Computational Thinking Practices

Design and Implement Creative Solutions and Artifacts: Students translate novel ideas into tangible forms. This design pattern encompasses steps of the problem solving cycle, including understanding, decomposing, and exploring the problem. Students create work products that show one or more designed solutions and/or artifacts, and then test and improve the solution and or artifact.

Apply Abstractions and Models: Students use ideas and representations that capture general to specific aspects or patterns of an entity or a process, and the coherent relationships/structures among entities or processes, including level-of-detail. These ideas and representations may move among different contexts (problem or disciplines). Students demonstrate knowledge of the representational properties of discrete mathematics.

Analyze one's own computational work and the work of others: Students demonstrate that they can understand problems, how they are solved computationally/algorithmically (including hardware), and how they may be unsuitable for a computational solution either by taking too many resources to solve or being unsolvable. Students show they can measure the accuracy, efficiency, and elegance of the solution.

http://pact.sri.com/

© 2012 Principled Assessment of Computational Thinking (PACT). Produced by the Center for Technology in Learning at SRI International with support from the National Science Foundation under contract numbers, CNS-1132232 and CNS-1240625 and CNS-0943507 to the University of Oregon. Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.