2018

Measuring Sophistication in Systemic Design and Computing

Barba, Evan and Osborn, J.R.

Measuring Sophistication in Systemic Design and Computing

Evan Barba, J.R. Osborn

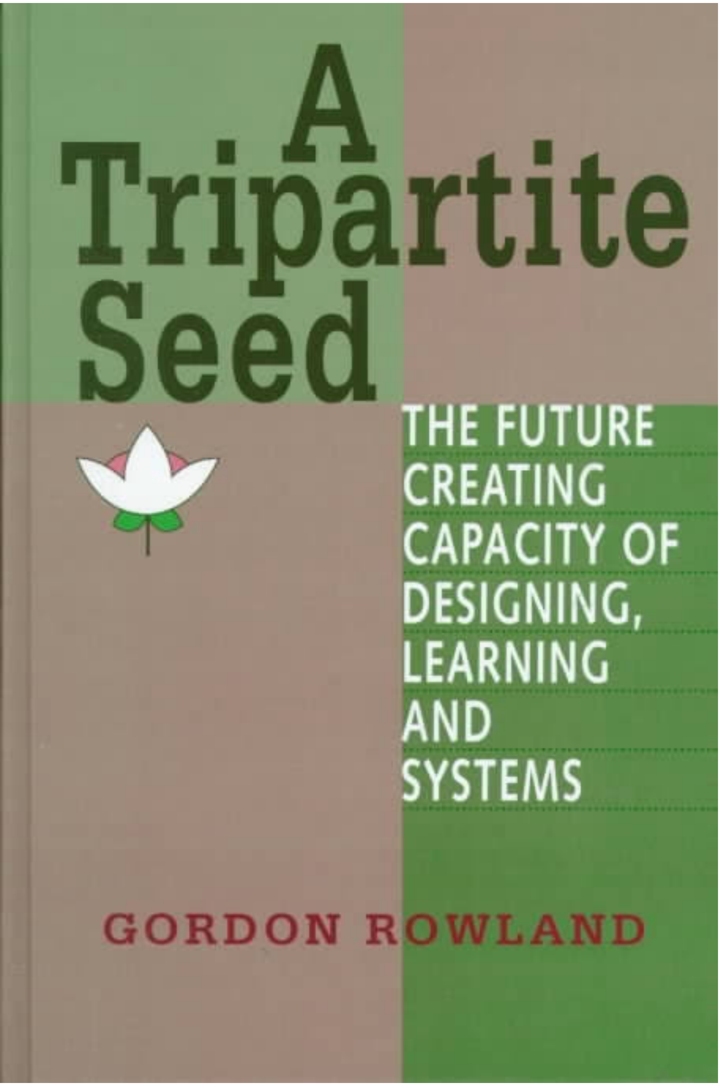
Communication, Culture and Technology
Program in Learning and Design
Department of Computer Science
Georgetown University, Washington, DC USA

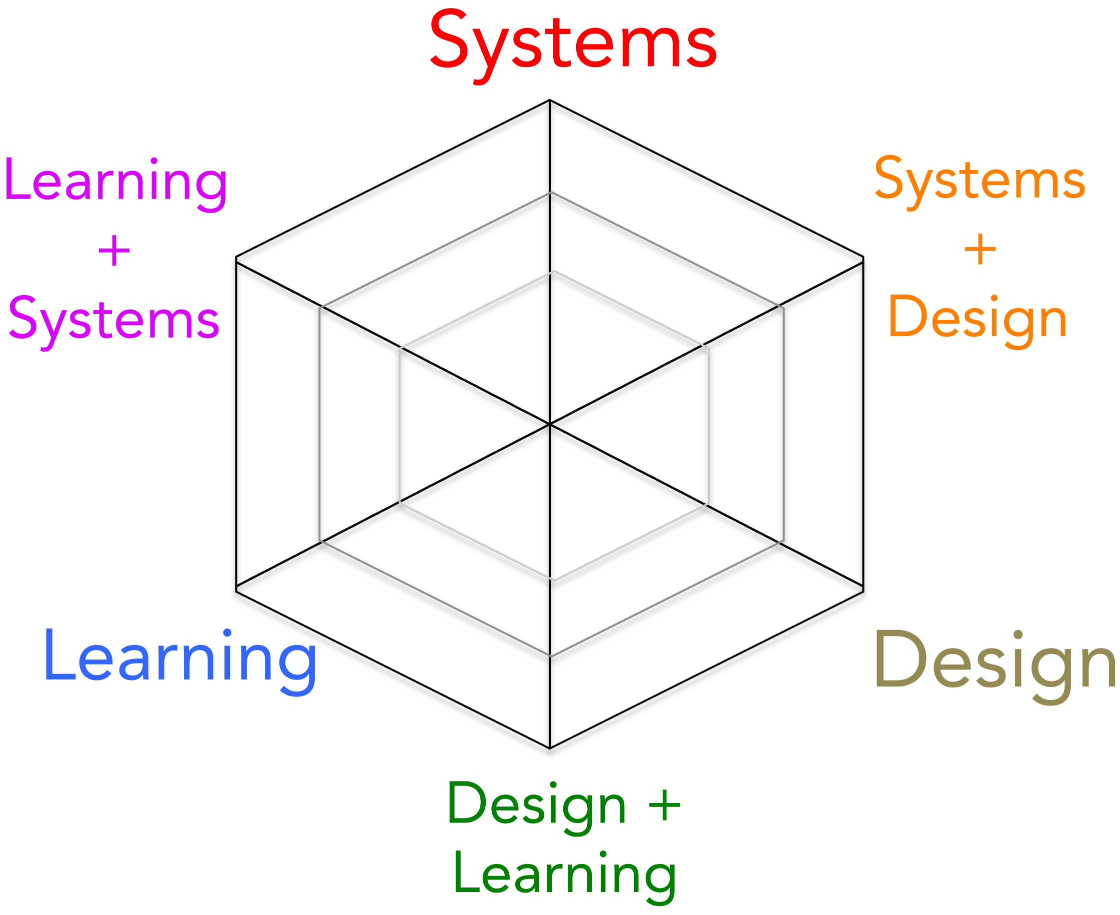
Assumptions

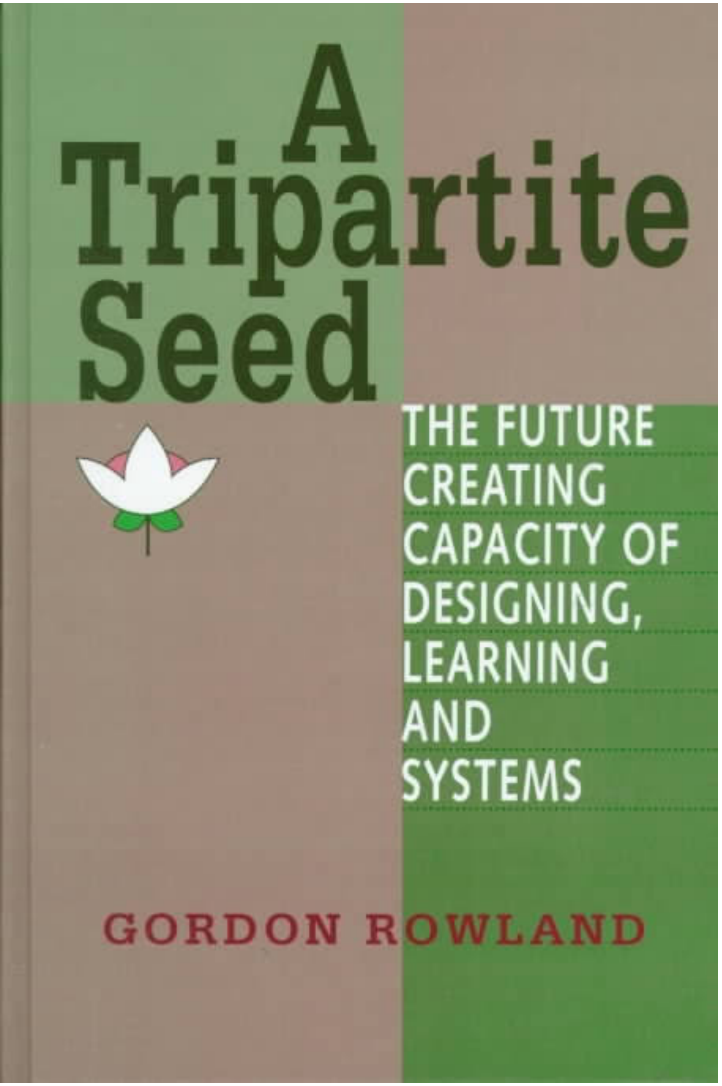
Learning to integrate disciplines is at least as important as learning a discipline

Interdisciplinary skills are *not* the same as disciplinary ones. But there is some overlap.

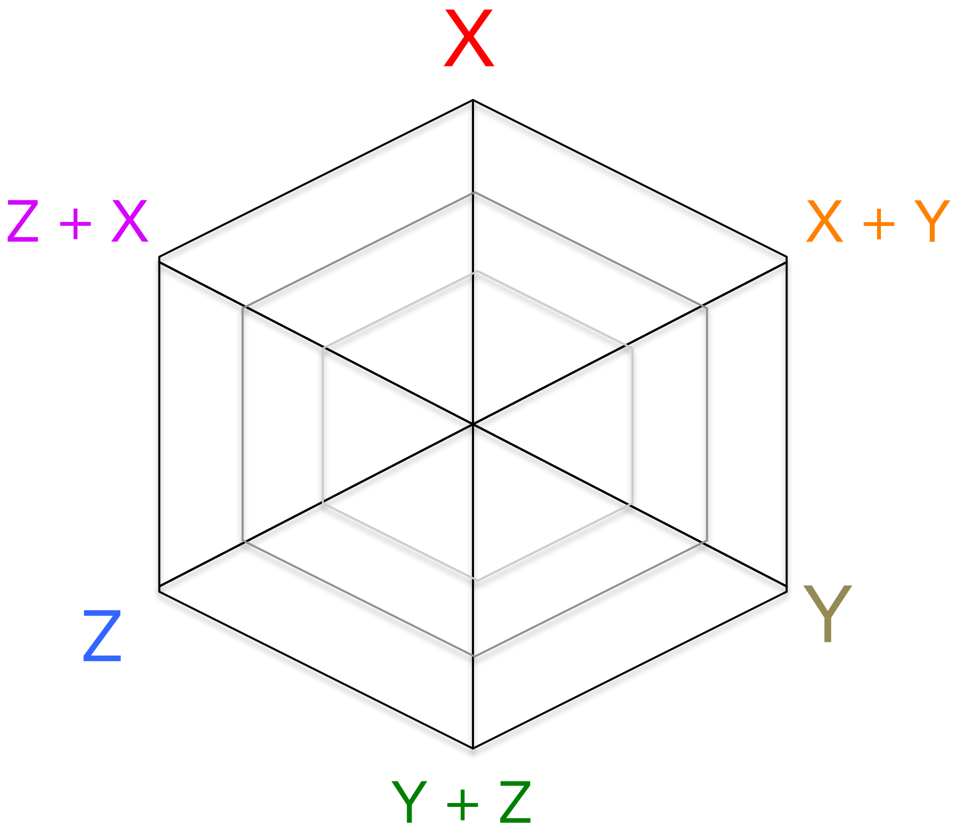
Objective metrics of learning are valuable

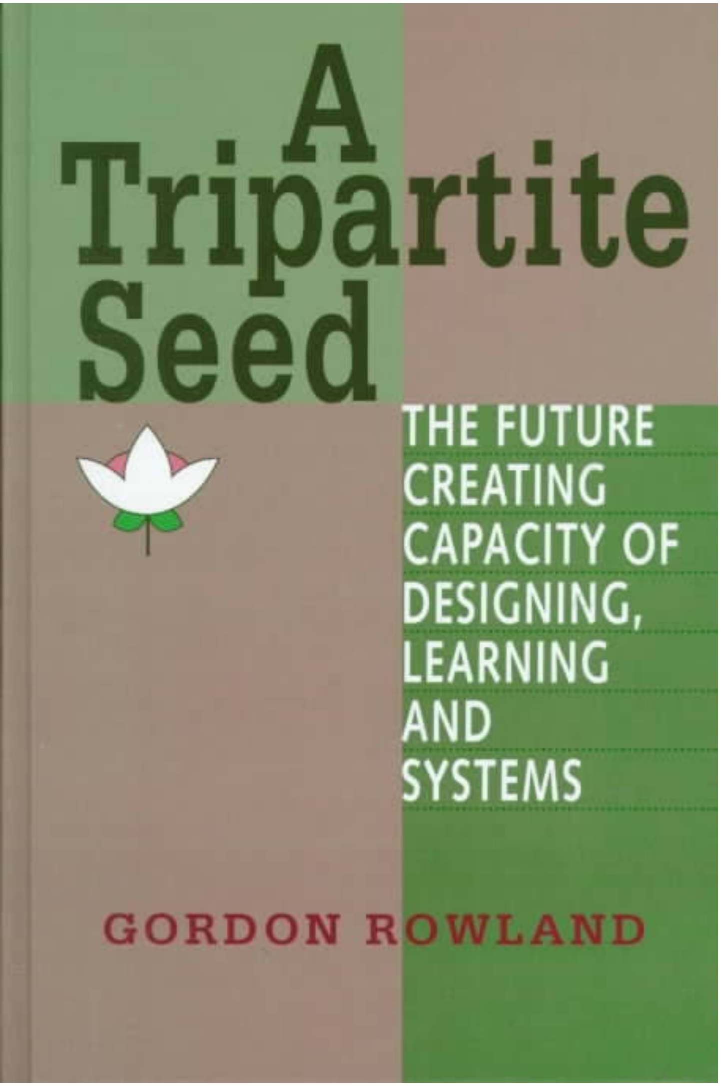

A Tripartite Seed
THE FUTURE CREATING CAPACITY OF DESIGNING, LEARNING AND SYSTEMS
GORDON ROWLAND


Systems
Systems + Design
Design
Design + Learning
Learning
Learning + Systems



This model can be applied to any three disciplines and intersections





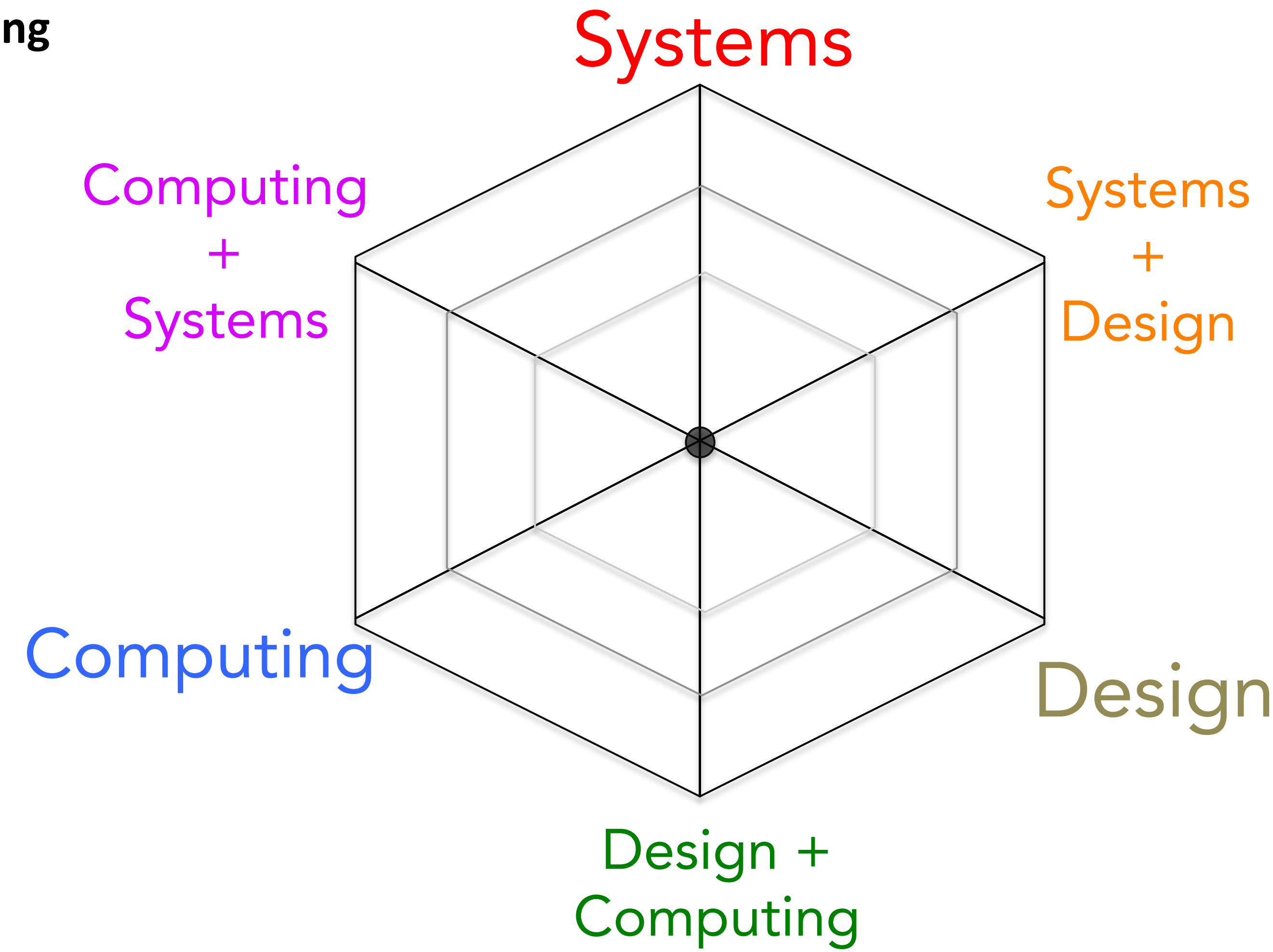
3 DISCIPLINES
&
3 INTERSECTIONS

More Disciplines & Intersections
becomes unwieldy

Systems as a Way of Thinking

Design as a Way of Working

Computing as a Medium



The ***Next Generation Science Standards*** identify seven "cross-cutting concepts" that:

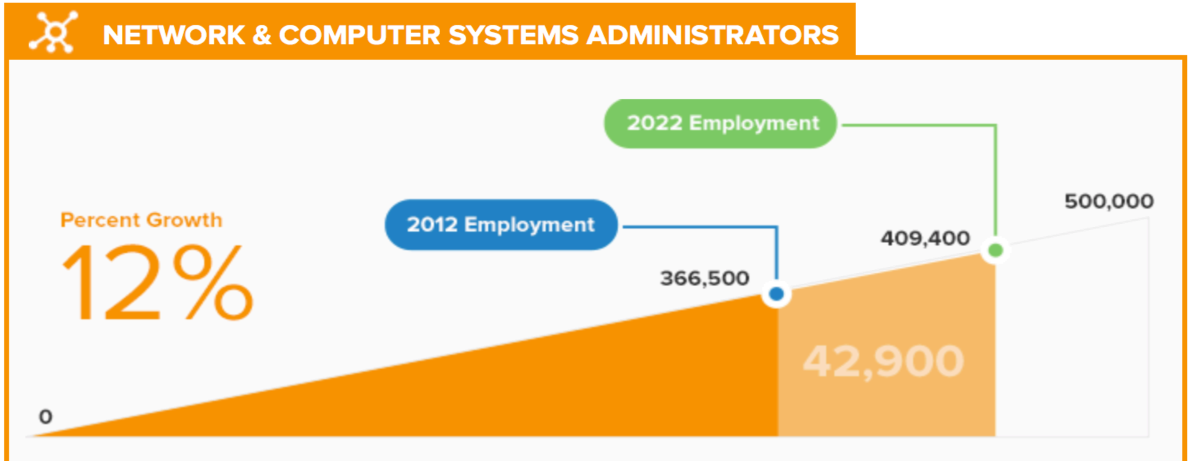
…need to be made explicit for students because they provide an organizational schema for interrelating knowledge from various science fields into a coherent and scientifically-based view of the world.

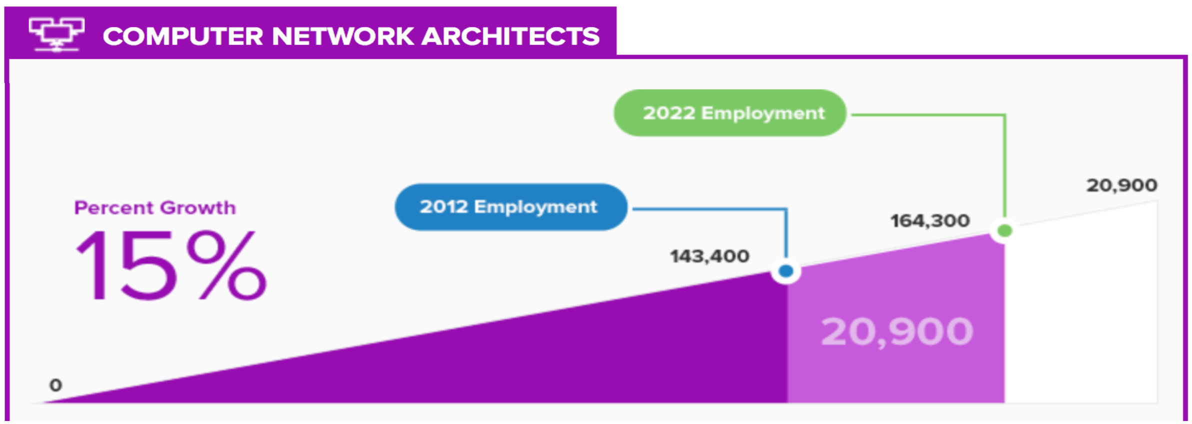
1. patterns
2. cause and effect
3. scale
4. system models
5. flows and cycles
6. structure and function relationships
7. stability and change.

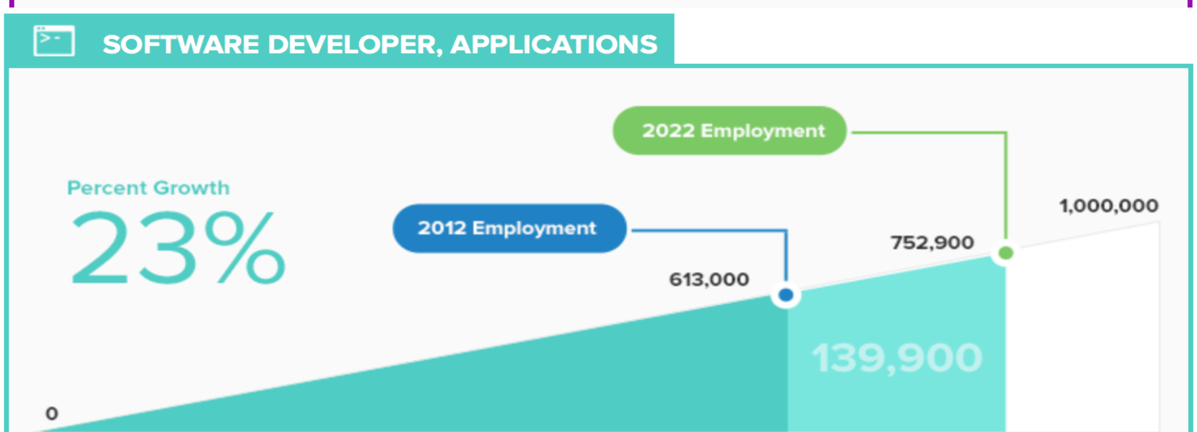
NGSS Lead States. 2013. *Next Generation Science Standards: For States, By States.* Washington, DC.

"…new liberal art of technological culture," (1992)

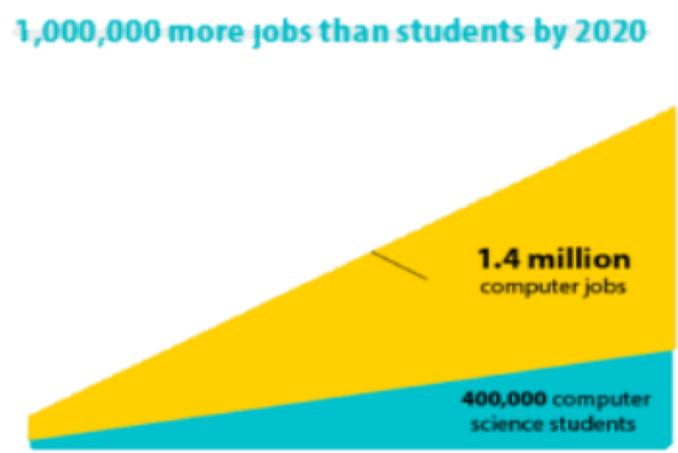
Richard Buchanan







Computing is where the job growth is



https://www.computerscienceonline.org/computer-engineering/

Learning Progressions

"Underlying any curriculum is a model of progression,"

M Hughes. 1996.

1. **Learning targets** that are defined by societal aspirations and analysis of the central concepts and themes *in a discipline*

2. **Progress variables** that identify the critical dimensions of understanding and skill that are being developed over time

3. **Levels of achievement** that define significant intermediate steps in conceptual/skill development

4. **Learning performances** which are indicative of skills and knowledge at each level, and which can be used in the development of assessments

5. **Assessments** that measure student understanding of the key concepts or practices and can track their progress over time.

Sophistication

"Learning is envisioned as a development of progressive sophistication in understanding and skills within a domain. [...] learning is conceived as a sequence or continuum of increasing expertise."

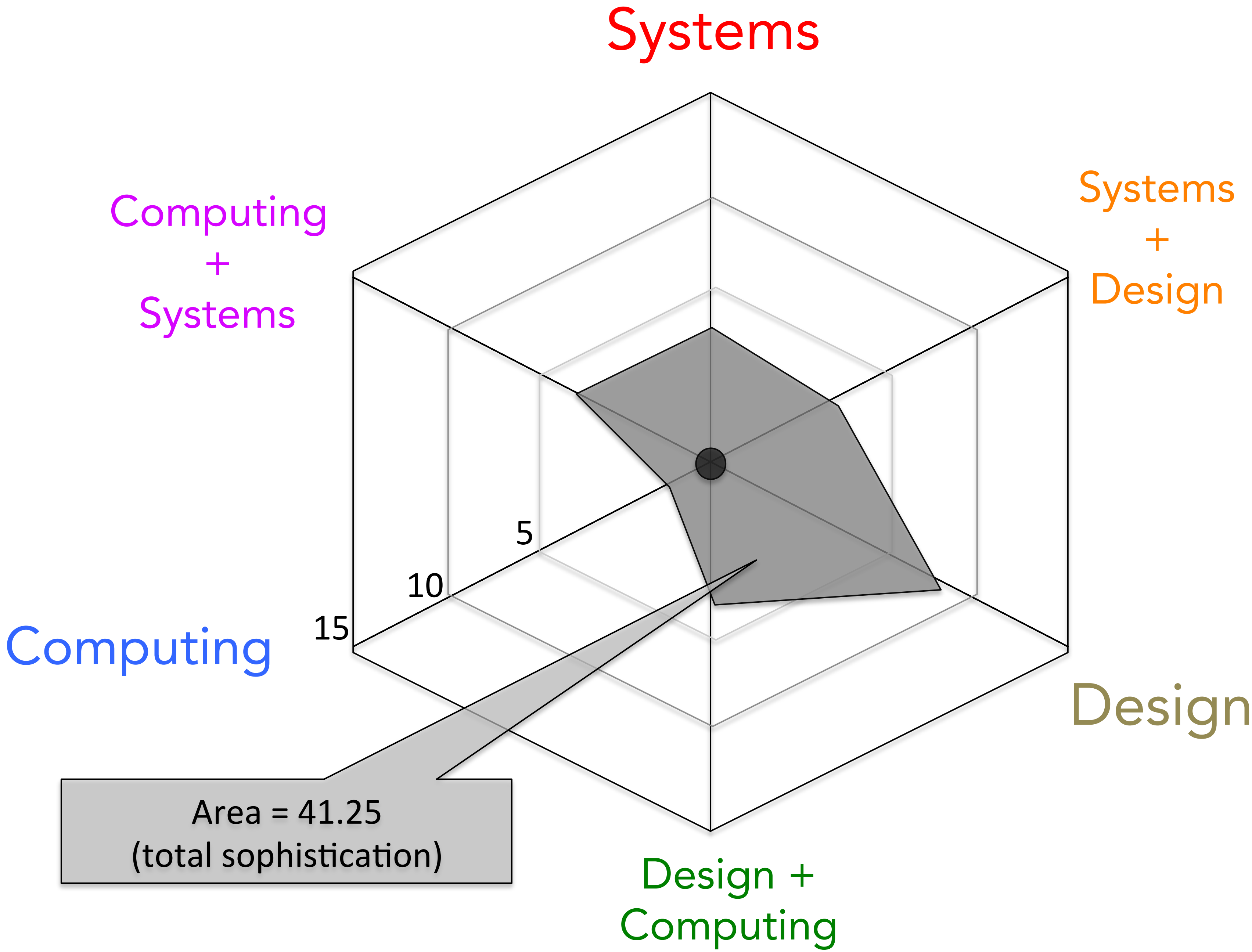
Heritage 2008

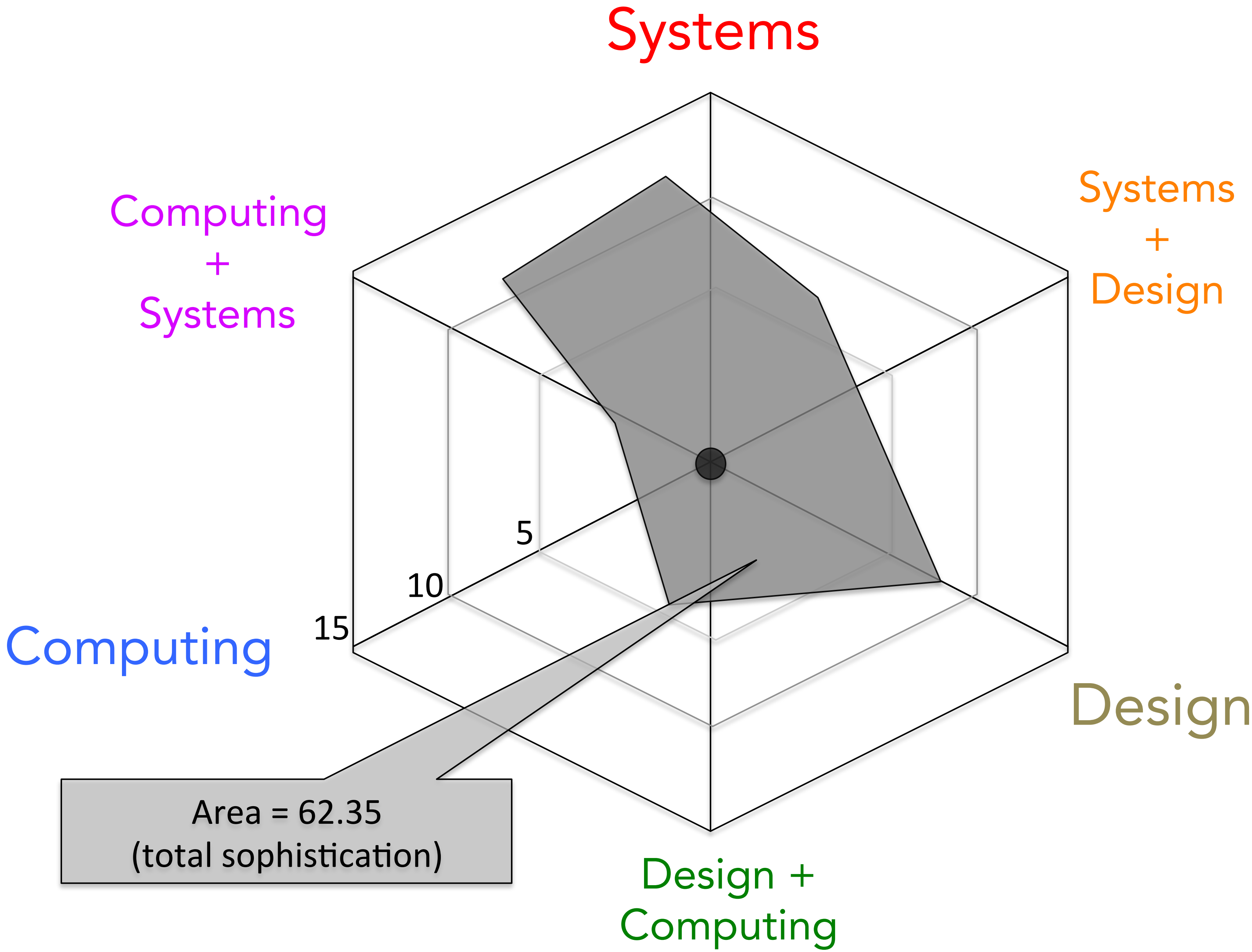
Criticisms

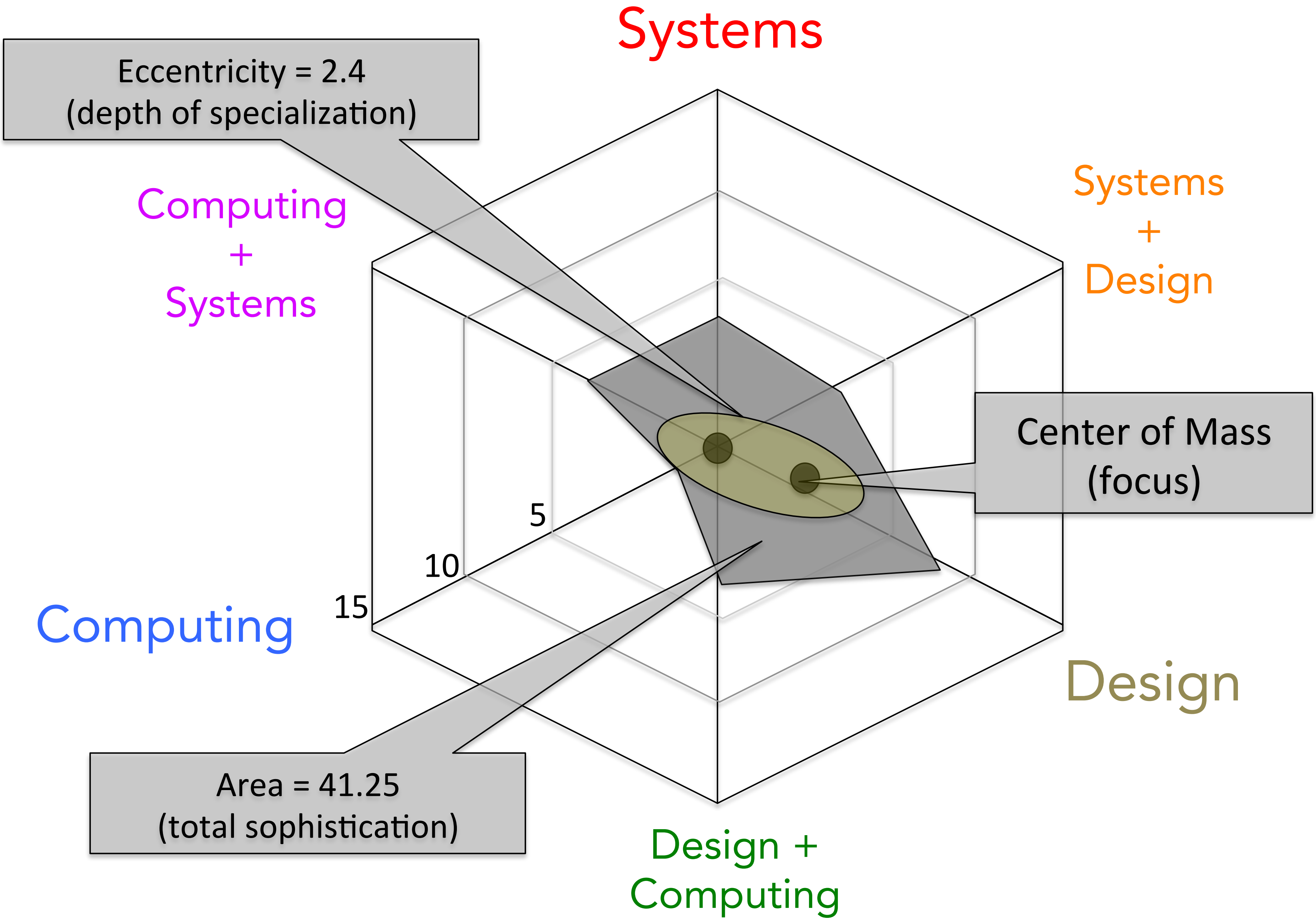
No accounting for errors, failures, false starts…

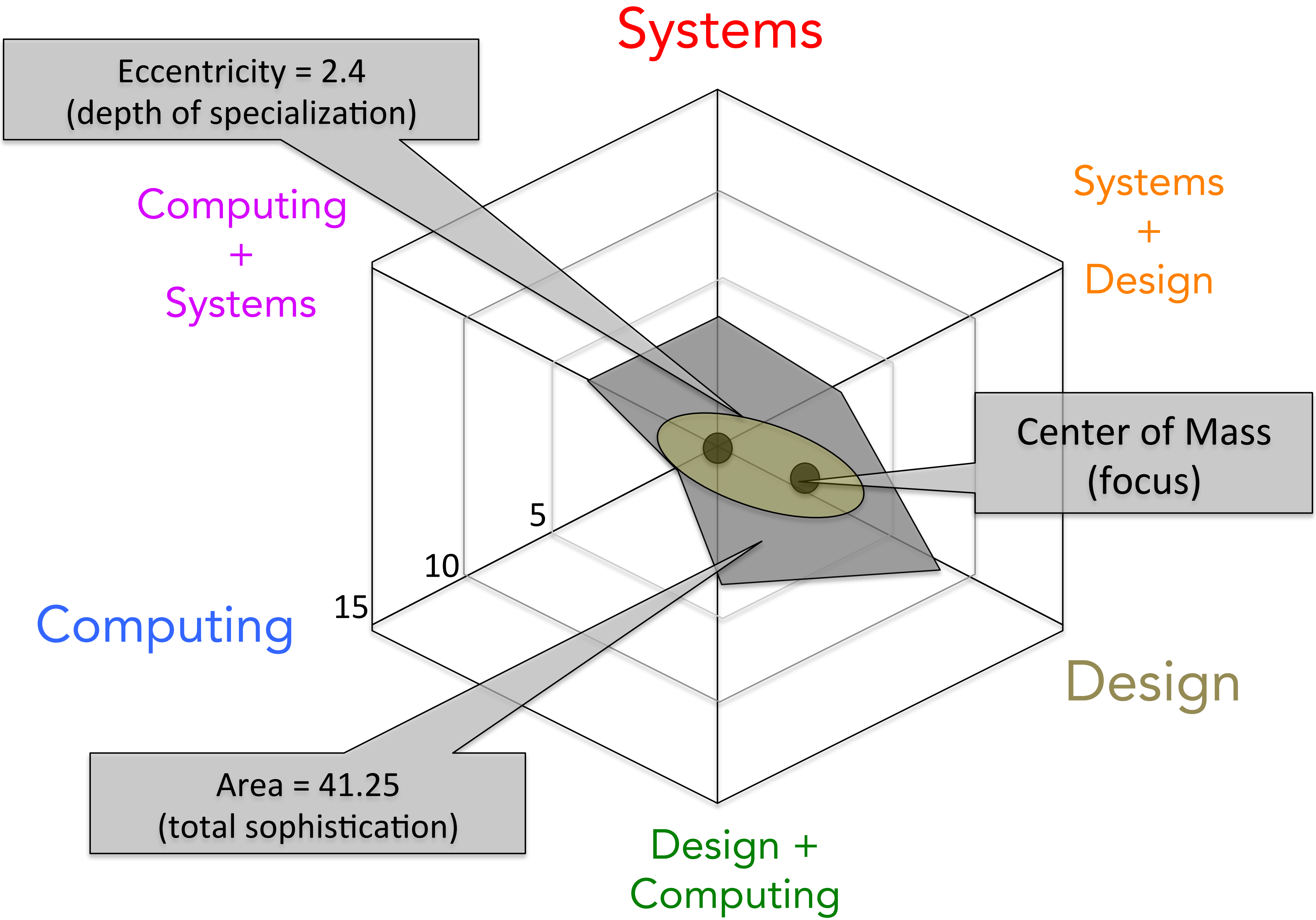
Not interdisciplinary

Assumes all learners are alike


Systems
Systems + Design
Design
Design + Computing
Computing
Computing + Systems
5
10
15
Area = 41.25
(total sophistication)


Systems
Systems
+
Design
Computing
+
Systems
5
10
15
Computing
Design
Area = 62.35
(total sophistication)
Design +
Computing


Systems
Eccentricity = 2.4
(depth of specialization)
Systems
+
Design
Computing
+
Systems
Center of Mass
(focus)
5
10
15
Computing
Design
Area = 41.25
(total sophistication)
Design +
Computing


Systems
Eccentricity = 2.4
(depth of specialization)
Systems
+
Design
Computing
+
Systems
Center of Mass
(focus)
5
10
15
Computing
Design
Area = 41.25
(total sophistication)
Design +
Computing

Advising

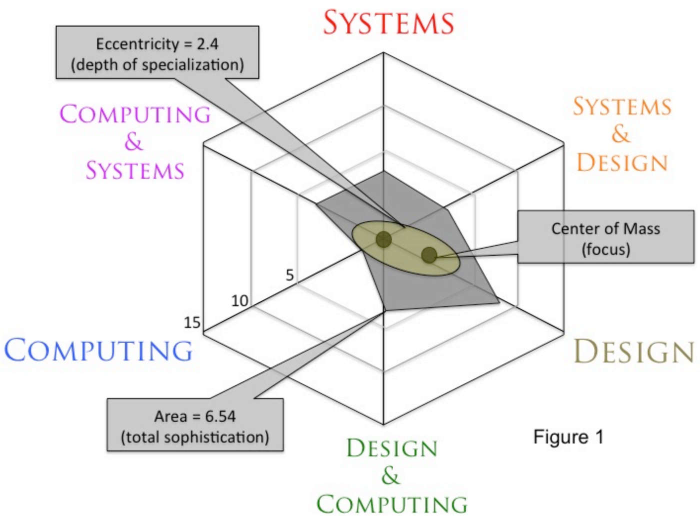


Image source: https://iris.peabody.vanderbilt.edu/module/tran-scp/cresource/q1/p01/

Table 1: Selection of levels defined for initial hypothesized learning progression

Subject, Progress Variable, Assessment	Construct Map	Progress Guide
Systems Progress Variable: System Mapping Learning Performance Category: System Map	1. Understands components and interactions and the distinction between system and environment; understands the basics of emergent and indirect effects 2. Can elaborate the relationships between components (stocks and flows, feedback loops, etc.); Can characterize and interrogate the interactions between the system and the environment 3. Can predict how a system might adapt to changes and indirect effects 4. Can apply systems concepts in new contexts to find insight or explain phenomena	1. The system is defined, but critical components, the environment, and interactions are missing 2. All relevant components and interactions are present, stocks and flows are labeled, and feedback loops are noted as postive or negative and given qualiative character; the depiction is straightforward and understandable 3. Perturbations and interactions with the environment are noted and labeled and time is considered as major factor and emergent effects are labeled, described or depicted 4. Scale and emergence are accounted for as are patterns of adaptation over time; the deptiction is complex but parsimonious ; multiple time-scales or perspectives might also be noted and depicted
Systems+Design Progress Variable: Intentional Emergence Learning Performance Category: Field (Delployment) Study/Design Plan	1. Identifies primary component (typically a user) and understands its interaction with other components of a system 2. Articulates indirect effects between the user and the system (i.e. constraints on the user imposed by the system and ways the user influences the system): adopts multiple perspectives in the design 3. can identify trade-offs between User-Centered and System-Centered approaches 4. Can identify emergent consequences of the intervention that affect both user and system 5. Can iterate to account for and optimize the observed emergent behaviors of both user and system	1. Both the user and system are described but the focus is on the immediate needs of the user; effects of the user on the system lack detail and do not unfold over time 2. The systemic constraints placed on the user (and the design) by the system are described in detail and the effects of the user on the system are clearly detailed 3. The conceived solution is deployable and shows evidence of tradeoffs needed to account for multiple perspectives 4. The observed behavior of deployed system is described in terms of both user and system effects 5. Iteration of solution makes appropriate trade-offs to optimize for both system and user
Design Progress Variable: Design Communication Learning Performance Category: Design Plans	1. Identifies opportunities for intervention and conceptualizes multiple solutions; likely gets fixated on one solution and cannot change course 2. Can create and follow a detailed plan resulting in a potentially deployable intervention; can communicate this plan at various points in multiple media; can adapt the solution partly 3. Reflects on and adapts the intervention during the design process as new contraints and opportunities arise; can adapt to outside feeback; complete the plan or prototype in a reasonable timeframe 4. Documents interim artifacts, and can recount rationale through every step of the design process; completes the project with enough tiime to add polish; has contingency plans and is flexible rather than fixed when changes are required.	1. Requirements gathering is done systematically although certain crucial elements might be overlooked; the solution seems sound; and diagrams, animations, slides, etc. are used to cleary exlpain how the proposed solution meets the observed needs; there is likely something crucial that was overlooked; sub-optimizes are explained away rather than adapted for 2. Critical flaws in the conceptualized plan are found and addressed rather than ignored; alternative solutions are explored 3. additional features of the design emerge to address previously unknown constraints or exploit new uses and opportunities 4. Documentation is robust and complete the rationale for the design and its evolition are clearly visible and explained well;
Design+Computing Progress Variable: Interaction Design Learning Performance Category: Prototyping	1. Can identify opportunies for interactive artifacts in a given context; can conceive of multiple assemblies of computational technologies that would be appropriate 2. Can specify the technical requirements for a given design and understands the limitiations of the technology in the context; can articulate the additional benefit adding technology would provide; 3. Can build a low-fidelity prototype; Can deploy and revise the prototype based on user feedback and observation; 4. Can add features that make the artifact robust to error and maintainable; documentaion or user guides are clear; interventions and artifacts can persist	1. Qualitavtive methods are used to understand a given environment and locate opportunities for design intervention; proposed intervetions are somewhat murky on details and not likely to be feasible due to poor understanding of the constraints 2. A feasilble intervention is put forth that is tuned in to the needs of the situation and appropriately scoped; there is still little sense or plan to implement the solution or systematically test it 3. A simple prototype is created to probe the intervention along the lines of important features adn this is used as the basis for iteration; there is a good sense of overall scope of the project 4.Equal attention is paid to user expectations and technical implementation; quick iteration is seen as essential for success; features are removed rather than added to enhance stability and simplify the experience
Computing Progress Variable: Programming Learning Performance Category: Programs and exams	1. Power User: can learn to use new tools, has an intuitive but naive sense for how data is represented and manipulated by these tools; can think through a problem in terms of logical steps and create a flow chart or similar representation 2. understands the core elements of a computer program (syntax, control flow, variables, methods, debugging); writes pseudocode 3. Can extend a simple program in a well-defined problem context; can locate logical errors and debug syntactial issues 4. Can implement a more complex program from a template; can formulate good questions when problems occur and seek out solutions from multiple sources but probably can't determine which are most useful; can debug logical errors 5. Can implement a complex program of their own design within a limited context; can locate external solutions and adapt them to their needs; Can work with existing code bases, define new compund data types and integrate with outside services	1. Student can learn to use new tools easily, anticipates results and can combine sequences of actions to achieve desired results; has minimal understanding of how data is being represented and manipulated 2. Identifies and defines programming elements in a given program; can locate syntactial errors; describes algorithms and data representations using correct but loose terminology, can implement a basic program 3. Simple program is functionally correct although spaghetti-like and overy complex; some features are likely not inpleemtned fully, but the core is there 4. Successful implementation of a more complex program that uses reusable methods and incorporates available widgets among other advanced techniques with some scaffolding 5. Program incorporates techniques that were not taught; can ask well-formulated questions using technical terms correctly; Program is written outside of a sandboxed environment and makes use of professional grade tools, services, and software packages that the student identifies themself
Computing+Systems Progress Variable: Modeling Learning Performance Category: Visualization/Simulation	1. Can identify variables appropriate for modeling in a given area of interest; Can organize variables and data by articulating relationships needed in the model 2. Can implement a small-scale sim/viz by choosing appropriate tools; looks for trade-offs that need to made to make the model more robost; starts to ask questions about what the sim/viz might tell us that we don't already know 3. Can implement a model of reasonable complexity and describe its features and limitations 4. Limitations of the implementation are clearly articulated; new insights can be drawn about the phenomena; and new extensions are conceived	1. Initial description has variables and relationships that are integral to the probelm, but are far too complex to model realistically; a sense of questions the simulation could answer is demonstrated, but it is overly ambitious 2. An appropriate question for the sim is asked and a proof of concept appears viable or at least appropriate flaws are detected; the sim runs and reveals a core relationship but no insight 3. A robust implementation is completed, and its limitations are being probed or articualted; revision of the initial questions is being considered; the sim/viz reveals some compelling behavior that can lead to insight 4. The implementation is expanded to include features that reveal new things about the phenomena, such as analytics or interactive elements to reorganize the data

Learning Progressions

"Underlying any curriculum is a model of progression,"

M Hughes. 1996.

1. **Learning targets** that are defined by societal aspirations and analysis of the central concepts and themes *in a discipline*

2. **Progress variables** that identify the critical dimensions of understanding and skill that are being developed over time

3. **Levels of achievement** that define significant intermediate steps in conceptual/skill development

4. **Learning performances** which are indicative of skills and knowledge at each level, and which can be used in the development of assessments

5. **Assessments** that measure student understanding of the key concepts or practices and can track their progress over time.

Questions