S2EA

System Thinking for Software Engineering Educators (and their students)

Prepared and presented by
Richard E. (Dick) Fairley, PhD
dickfairley@gmail.com

S2EA

A brief bio

Richard E (Dick) Fairley, PhD is principal associate of Software and Systems Engineering Associates (S2EA), a consulting and training company. He is also an adjunct professor of computer science, software engineering and IT in the online doctoral program at Colorado Technical University. He was an author of the Guide to the Systems Engineering Body of Knowledge (SEBoK) and is currently the co-editor of SEBoK, Part 7. He was an author and co-editor and of the current version of the Guide to the Software Engineering Body of Knowledge (SWEBOK V3) and was lead author and editor of the software engineering competency model (SWECOM). He was also leader of the team that developed the Software Extension to the PMBOK® Guide. Dr. Fairley has bachelors and masters degrees in electrical engineering and a PhD in computer science and applied mathematics. Dick and his wife Mary Jane reside in Teller County, Colorado – their home is at 9000 feet elevation. His avocations are motorcycling in summer, skiing in winter, and is currently on sabbatical from hosting a jazz show as DJ on Sunday evenings on KRCC radio year-round (streamed on KRCC.org).

2

S2EA

My recent-past professional activities

- Chair, IEEE-CS Software and Systems Engineering Committee (2010 to 2015)
- Co-chair, co-editor, and an author of SWEBOK V3
- Leader of the SWECOM team
  - The Software Engineering Competency Model
- Leader of the CS-PMI SWX team
  - Software Extension to the PMBOK® Guide (SWX)
- An author of SEBoK
- A past ABET CAC Commissioner
- Author: "Northwest Hydro System" *SEBoK Part 7*
- Co-author: "Managing Technical Debt in System Development" *IEEE Computer*

3

S2EA

URLs

- SWEBOK: https://www.computer.org/web/swebok/v3
- SWECOM: https://www.computer.org/web/peb/swecom
- SWX: http://marketplace.pmi.org/Pages/ProductDetail.aspx?GMProduct=00101457501
- SEBoK: sebokwiki.org
- BKCASE: bkcase.org
- "Northwest Hydro System": (SEBoK Part 7) http://sebokwiki.org/wiki/Case_Studies
- "Managing Technical Debt": http://ieeexplore.ieee.org/document/7924273/

4

S2EA

My current professional activities

- Editor: Part 7 of SEBoK (The Guide to the System Engineering Body of Knowledge)
- Team Member: ABET accreditation criteria for cyber-security engineering programs
- Active participant of INCOSE, IEEE, and PMI
- In progress:
  - "A systems approach for Enterprise Information Systems"
  - *System Engineering for Cyber-Physical and Software-Intensive Systems*

5

S2EA

Presentation agenda

- What is a system?
- What is system thinking?
- System thinking inhibitors
- Some antidotes

6

S2EA

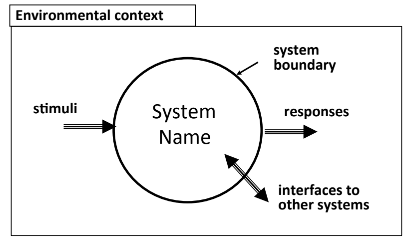
What is a system?

- A system is a collection of interconnected and interacting elements that exist within and interact with an environment
- A system may include some or all of:
  - Natural elements (e.g., wind, sun, water)
  - Physical-engineered elements (e.g., wind farms, …)
  - Software-engineered elements (infrastructure, applications)
  - Human elements (e.g., operators, maintainers)
- The environment includes similar kinds of elements
  - The humans are users and other stakeholders

7

S2EA

Context and environment



Q: What's inside the system and what's outside?

8

S2EA

Two kinds of systems

1. Cyber-physical systems
   - Physical elements must be fabricated and/or procured
   - Software must be constructed and reused
2. Software-intensive systems
   - The computing platform and infrastructure are available
   - Any needed physical elements are available

9

S2EA

Presentation agenda

- What is a system?
- What is system thinking?
- System thinking inhibitors
- Some antidotes

10

S2EA

What is system thinking?

- System thinking involves adopting a *holistic mindset* to understand:
  - the scope of a system;
  - the system's context and environment;
  - relationships to affiliated systems;
  - functionality, behaviors, and quality attributes provided by (or to be provided by) the system.

11

S2EA

Four keywords of system thinking

1. Holism
2. Context
3. Emergent behaviors
4. Unintended consequences

12

S2EA

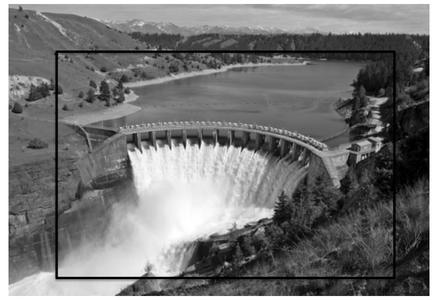
Holism

- **Holism:** A complex system cannot be fully understood by understanding the separate parts; the entire system and the environment in which it exists must be examined
- Some examples:
  - Air traffic control
  - Health care systems
  - Modern automobiles
  - Enterprise information systems
  - Hydroelectric dams

13

S2EA

A cyber-physical system



http://bloximages.chicago2.vip.townnews.com/missoulian.com/content/tncms/assets/v3/editorial/9/06/90637af4-56b2-11e2-a514-0019bb2963f4/50e744a759034.image.jpg

14

14

S2EA

Physical elements of a hydroelectric dam

Natural elements
- Reservoir
- Gravity
- Water flow

Engineered elements
- Penstocks
- Spillway
- Turbines
- Generators
- Step-up transformers
- Transmission lines
- Substations
- Salmon ladders

The natural and engineered elements are sensed and controlled by software

15

S2EA

Hydro-dam engineering disciplines

- Civil engineering
- Mechanical engineering
- Electrical engineering
- Software engineering
- Ergonomics engineering
- Hydrological engineering
- Environmental engineering
- Physical security engineering
- Information security engineering
- Safety engineering

Most, if not all of these disciplines will incorporate software elements

16

S2EA

CPS software

- Software in a CPS senses, measures, regulates, and controls the physical elements
  - The software is in embedded digital devices that are invisible to the system users – hardware, software, humans
- The software elements of a CPS:
  - Provide interconnections among the physical elements;
  - Coordinate behavior of the physical elements;
  - Provide interfaces to the external environment; and
  - Provide significant functionality and behavior.

17

S2EA

Software-intensive systems

- A software-intensive system is one for which software is the primary focus of a system development or modification project
  - To be implemented on a pre-existing platform of hardware and software
  - e.g., a hydro dam control room

Any software system where the platform and infrastructure are predefined

18

S2EA

A software-intensive system

http://www.energent.com/tag/bc-hydro/

19

S2EA

Context

- **Context:** understanding the system environment and the ways in which a system is intended to be used supports understanding of required and desired system functionality, behaviors, and quality attributes
  - Consider self-driving vehicles
  - Or hydroelectric dams

20

S2EA

Emergent Behaviors and Unintended Consequences

- **Emergent behaviors:** new, unanticipated system behaviors may emerge as system elements are combined or modified
- **Unintended consequences:** emergence may result in unforeseen, and often undesired, effects within the system or on the system environment

21

S2EA

Impact of legacy elements

- User interfaces for new and legacy elements may be incompatible
- Bridges and gateways may be needed to provide compatible user interfaces, internal interfaces, and data formats
- Excessive hardware and software resources may be needed to support unwanted and unneeded legacy features
- Some undesired legacy features may have to be masked
- Emergent behaviors: new unwanted and undesired features may emerge when legacy elements, tailored elements, and new elements are included
- Unintended consequences: some undesired behaviors may not be immediately evident

22

S2EA

Presentation agenda

- What is a system?
- What is system thinking?
- System thinking inhibitors
- Some antidotes

23

S2EA

System thinking inhibitors

Four inhibitors:

1. Illiteracy
2. Terminology confusion
3. Approaches to problem solving
4. Current process models

24

S2EA

1. Illiteracy

1. Many "SwEs" do not have CS or SwE backgrounds
2. Many SwEs do not belong to a professional organization, read SwE literature, or attend professional meetings
3. Some software engineers are narrowly focused detail-oriented specialists

   And have no interest in anything outside of their narrow focus

S2EA

2. Terminology confusion

- Agile
- Increment
- Iterative
- Verification
- Validation
- Quality assurance
- System maintenance
- Object oriented
- Fabricate vs. construct

What do these terms mean to different SwEs and others?

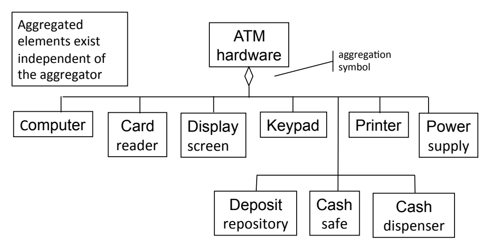
S2EA

3. Approaches to problem solving

- Traditional engineers focus on the problem domain
  - and the physical elements needed to solve the problem
  - With some supporting software
- SwEs focus on the solution domain
  - and model software elements on domain elements
  - with supporting infrastructure

S2EA

Aggregation of physical system elements

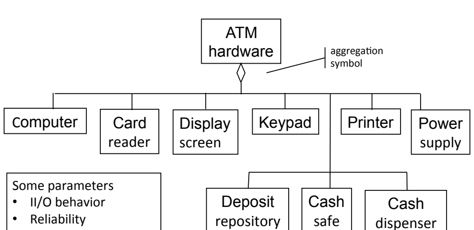


Nine system elements: each satisfies a set of requirements and can be independently fabricated or procured

Aggregated elements are "black boxes"

S2EA

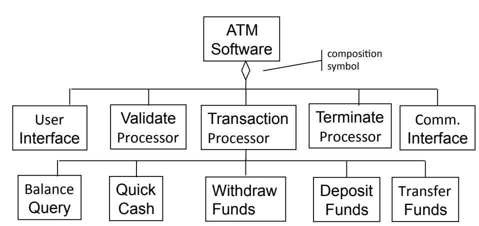
A "hardware first" SysE solution



Some parameters
- II/O behavior
- Reliability
- Mass
- Volume
- Power consumption
- Heat generation
- Interface connections

S2EA

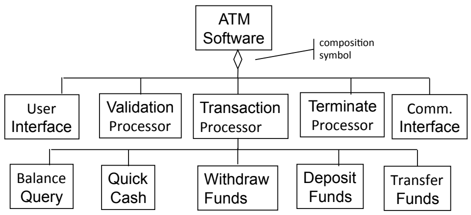
Composition of software elements



Instantiations of the composed elements exist only during instantiation of the composing element

S2EA

A "software first" SwE solution



- Each module provides a *service* for ATM customers
- Each class definition is a parameterized "black box" that can be tailored and instantiated

31

S2EA

An observation 1

- "Hardware first" and "software first" development strategies are the primary reason systems are not completed on time and/or are inadequate

32

S2EA

An observation 2

- Either the software is late (hardware first strategy)
  - "we were waiting for the hardware folks to tell us what software they needed"
- Or the hardware is late (software first strategy)
  - "we were waiting for the software folks to tell us what hardware they needed"
- And the hardware or software or both may lack adequate features, functionality, behavior, and/or quality attributes

33

S2EA

System thinking inhibitors

Four inhibitors:

1. Illiteracy
2. Terminology confusion
3. Approaches to problem solving
4. Current process models

34

S2EA

The problem is

- Traditional engineers prefer incremental hardware development models - sequences of mini-waterfalls
  - With increments that may span two or three months or longer
    - To fabricate and/or procure system elements
- Software engineers prefer iterative software development models
  - With iterations that span a couple of weeks
    - To construct and adapt system elements

35

S2EA

The solution is

- A development process that supports collaborative development of prioritized *capabilities*
  - with system increments that may span two or three months
  - And software iterations that typically span a couple of weeks
    - Embedded in system increments

36

S2EA

A capabilities-based approach

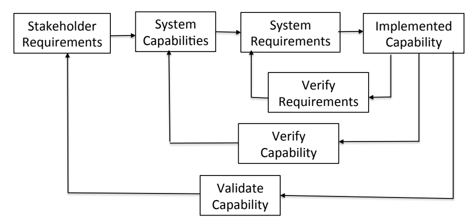
- A capability is a system feature, a behavior, and/or a quality attribute
  - o that requires some or all of hardware, software, and manual elements to implement
  - o Individual capabilities can be grouped into prioritized capability sets
- System capabilities are derived from stakeholder requirements
  - Quantified technical requirements are derived for the capabilities

Hardware, software, and manual requirements should not be allocated separately

37

S2EA

Capabilities implementation



38

S2EA

A stakeholder requirement

"Users want to obtain cash from ATMs at convenient times and locations,

and to do so in a safe and secure manner"

39

S2EA

ATM system capabilities

System capabilities will include:

- o An authentication mechanism
- o Secure access to financial accounts
- o Various kinds of financial transactions
- o Real-time responsiveness
- o Time-stamped transaction logs
- o Printed receipts of transactions
- o Information security
- o Physical safety and security

Top-level capabilities are decomposed into subordinate capabilities

40

S2EA

Hardware/software support for capabilities

Hardware/ software	Balance Query	Quick Cash	Withdraw Funds	Deposit Funds	Transfer Funds
Screen	X	X	X	X	X
Keypad			X		X
Safe		X	X		
Dispenser		X	X		
Repository				X	
Printer	X	X	X	X	X

- What combination of hardware and software is need for each capability?

41

S2EA

The I^3 development model

- I^3: Integrated Iterative-Incremental
- I^3 integrates the incremental approach preferred by system engineers and the iterative approach preferred by software engineers
  - o To incrementally develop prioritized system capabilities

42

S2EA

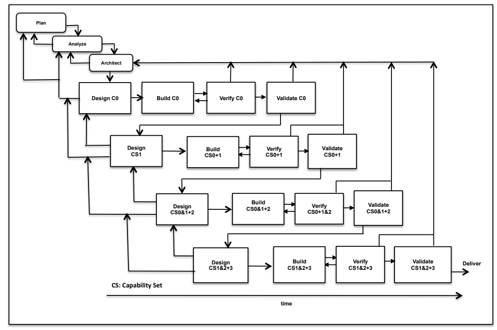
An MBSE Approach

- First build a software simulation of the architectural skeleton of the physical system
- Select one of more prioritized capabilities for implementation
- Software engineers develop software elements for the capabilities using simulated hardware interfaces
- System engineer and disciplinary engineers systematically replace simulated physical elements with real physical elements as they become available

43

S2EA

The I^3 development model



44

S2EA

Three objections to the I^3 model

1. The "overhead" of building C0
2. The "overhead" of elaborating and refining the evolving I^3 architectural model
3. The "overhead" of establishing and maintaining the integrity of the evolving I^3 architectural model

45

S2EA

Response

A well-known, common sense rule–of-thumb:

work performed "up-front" pays big dividends downstream

46

S2EA

Presentation agenda

- What is a system?
- What is system thinking?
- System thinking inhibitors
- Some antidotes

47

S2EA

Some antidotes

For:

1. Illiteracy
2. Terminology confusion
3. Approaches to problem solving
4. Current process models

48

S2EA
Illiteracy and Terminology antidotes - 1

- Seminars
- Invited talks
- Interdisciplinary classes and projects
- Internships
- Student subscriptions

S2EA
Illiteracy and Terminology antidotes - 2

A glossary for each class:
- SwE terms
- Domain-specific terms
- Project-specific terms
- Exert baseline control
  - publicize additions and changes
- Initial preparation and updating is coached by the instructor
- Glossaries are class deliverables

See ISO/IEC/IEEE Standard 24765:2010 Aka "SEVOCAB"
https://pascal.computer.org/sev_display/index.action

S2EA
Problem solving antidotes

- Require a term project for each class
  - With assigned teams of 3 students
  - And weekly deliverables that culminate in a final report
- Intermediate deliverables are reviewed and returned to each team for revision, if needed
- Note: projects must require some system thinking to include consideration of the context, environment, and impact of the result

Teams are coached by the instructor

S2EA
Project assignments and project courses

- Require weekly face-to-face weekly meetings with meeting agendas
  - To review progress and assign tasks
  - And coaching by the instructor
- It is a mistake to let project assignment and project course be "self-guided" and "self-managed"
  - Emphasize an environment for learning about leadership and teamwork
    - Require rotating student leadership
      - With instructor coaching and supervision

S2EA
Project courses

- Teach and emphasize a capabilities-based approach
  - User requirement to capabilities to system requirements with traceability
- Use an increment-iterative process
  - Software simulation of hardware and environment interfaces
    - Based on a system architecture
  - Iterative development of the software elements
    - With frequent periodic demonstrations of progress
    - And revision of the external interfaces as needed

With instructor coaching and supervision

S2EA
Use activity-based learning

- Tell me and I forget, teach me and I may remember, involve me and I learn.

  -- Benjamin Franklin

- Action Learning is involves small groups working on real problems, taking action, and learning as individuals and as a teams

  https://wial.org/action-learning/

S2EA

Action Learning approach

- This approach emphasizes discussion and interaction during class time
  - To replace lecturing
- Supported by a learning management system (LMS)
- Requires advanced instructor preparation
  - That is repaid by decreased workload during the class
    - And increased satisfaction for the instructor
  - Provides increased student engagement
  - Allows reuse of learning materials
- Also helps to reduce the problems of u/g project teams

55

S2EA

Learning management systems

An LMS can be used to organize and post, for each course:

- Syllabus
- Rubrics
- Lecture notes
- Reading assignments
- Discussion questions
- Assignment templates
- Grade book

By class session or by week

56

S2EA

Presentation agenda

- What is a system?
- What is system thinking?
- System thinking inhibitors
- Some antidotes

57

S2EA

Comments?
Questions?

58