Systematic Innovation for Computing

David W. Conley

INNOMATION CORP
Systematic Technology & Business Innovation

Outline

Most organizations that struggle with consistently producing coding that meets their customer's expectations mistakenly believe that their programmers simply need improved syntax creation skills.

Outline

CMMI (what does it tell us?)
- Process Maturity & Organizational Maturity
- Need for Standardized System Level Processes

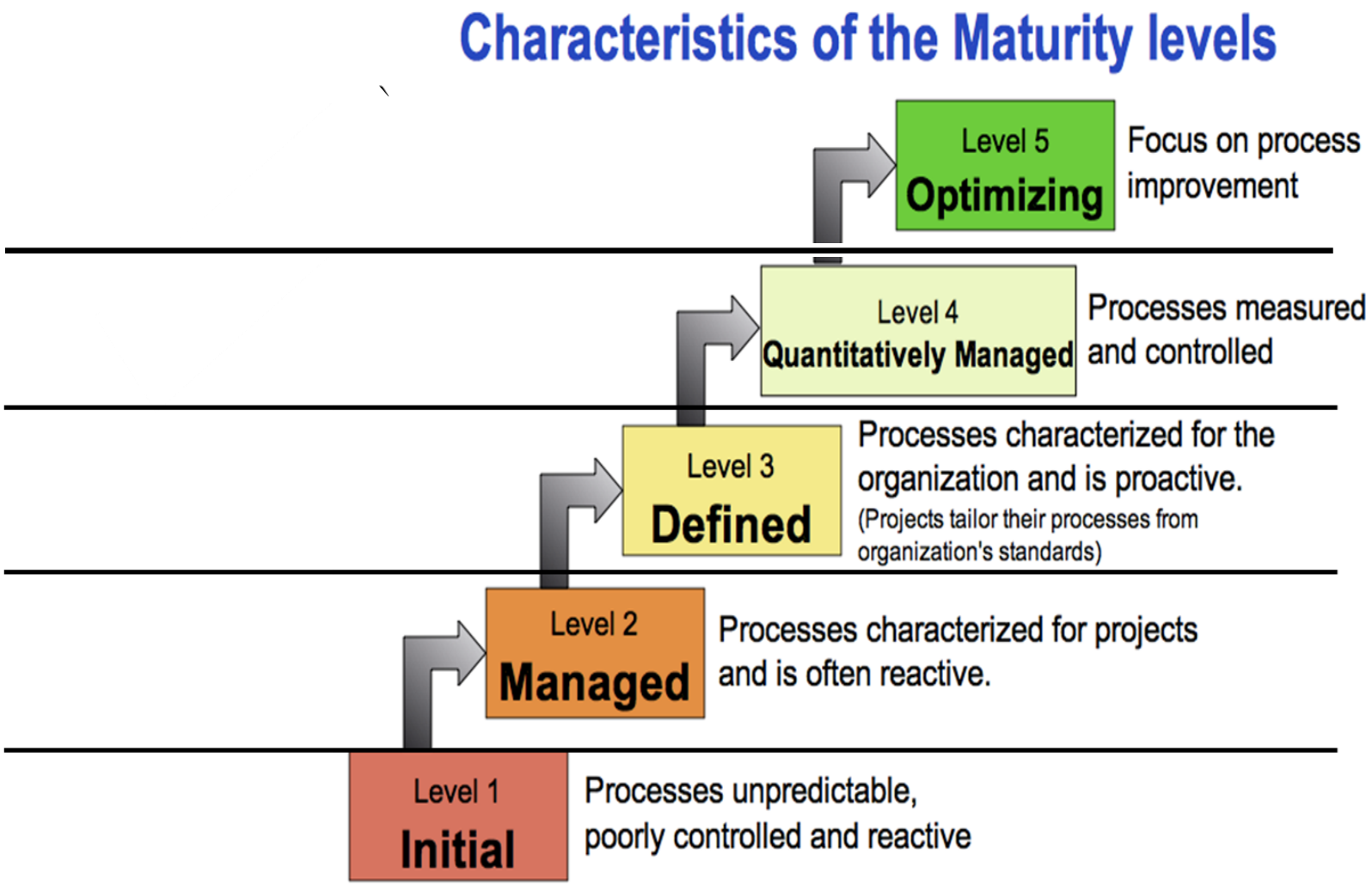
Systems Engineering Analysis
- Functional Model
- Analysis Direction
- Concentric Hierarchy of Control System
- Coding Modules

Additional Techniques
- "Informs" Resolution
- Resource Analysis

Summary

Capability Maturity Model Integration (CMMI) – What does it tell us?

Organization Characteristic Rankings



Capability Maturity Model Integration (CMMI) – What does it tell us?

Highly ranked CMMI organizations consistently produce higher quality and more robust software.

However, programming skill is not a category in CMMI rankings.

Why do high level CMMI orgs produce better software?

Process and Operations Maturity substantially based on systems level analysis and planning

How can any organization start their improvement journey?

Systems Level Analysis

Systems Analysis

"Systems analysis is the study of sets of interacting entities, including computer systems analysis."*

According to the Merriam-Webster dictionary, systems analysis is "the process of studying a procedure or business in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way."

Yet most programmers create coding without fully understanding the engineering system for which they are coding.

*** Albus, James S. (2014, May 27th). *Systems Analysis*. Retrieved from http://en.wikipedia.org/wiki/Systems_analysis**

Systems Analysis

Systems Analysis can be applied to technical systems or processes

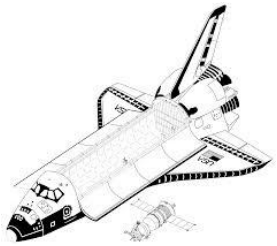
Systems Engineering Analysis –

- Understand the technical system being supported
- Insure the design supports the technical system
- Insure the design does not cause unexpected effects

Process Analysis –

- Understand the process being used to create the product
- Insure the process supports the product development
- Insure the process does not cause unexpected results

Systems Engineering Analysis

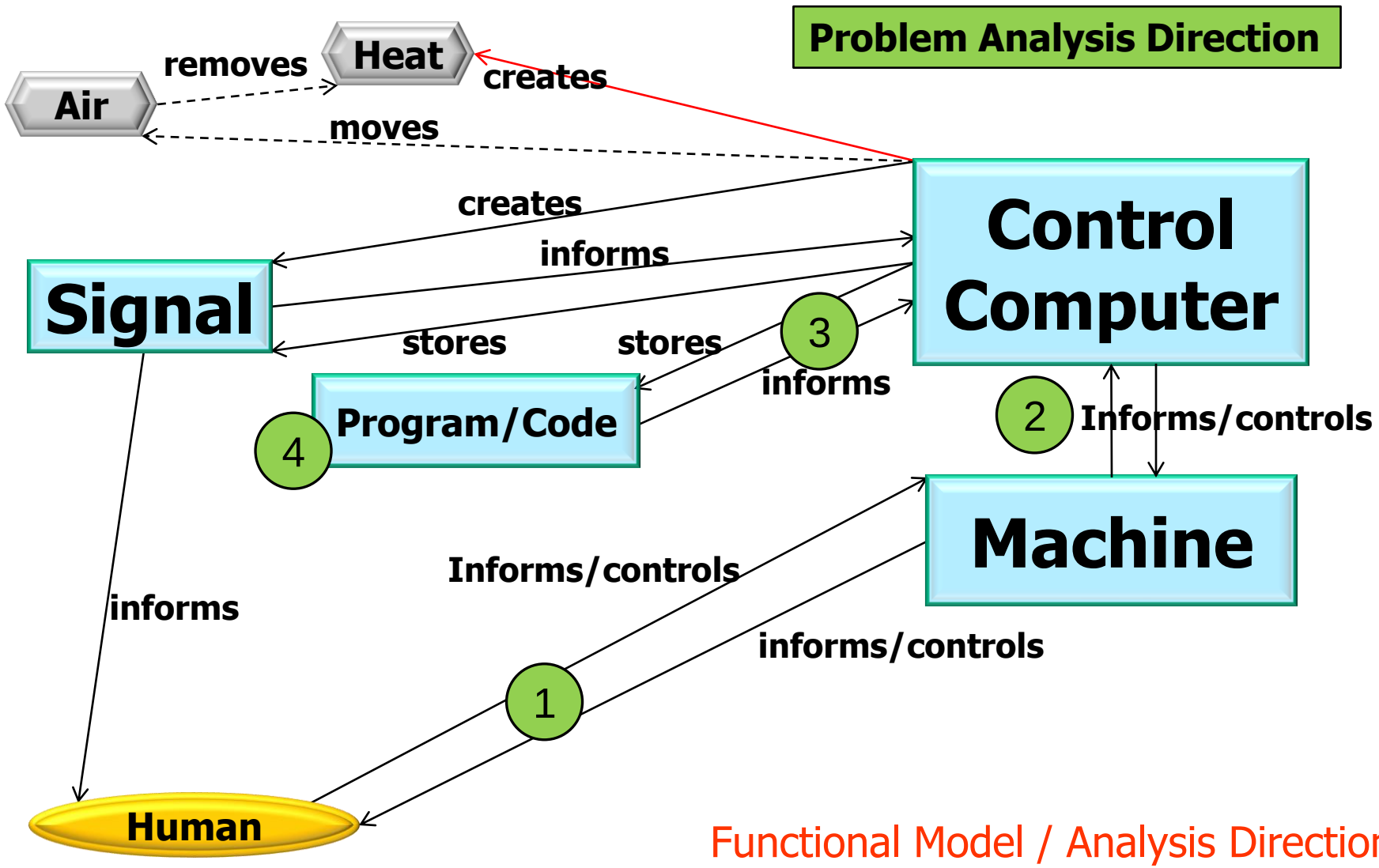


Most computing systems are put in place as control systems for a larger "physical" engineering system

Computing system improvement should not focus on improving the software /coding but rather the systems' main function

Therefore analysis of software /coding should always analyze the coding in context of the larger engineering system

Systems Engineering Analysis



Systems Engineering Analysis

Problem Analysis Direction

step 1
- Machine/Human level
- Machine/Super System level

How can the interaction between the machine and the super system (or human) be improved?

step 2
- Machine/Computer level

Does the improvement idea from step one require improvements between the machine and the control computer?

Could the operation of the machine (within itself) be improved by improving the interaction between the machine and the control computer?

step 3
- Computer/Software level

Does the improvement idea from step one or step two require improvements between the control computer and the software?

Could the operation of the computer (within itself) be improved by improving the interaction between the control computer and the software?

step 4
- Inter-Software level

Does the improvement idea from step one, step two or step three require improvements within the software?

Could the operation of the software be improved by improving the interaction within the software itself?

Functional Model / Analysis Direction

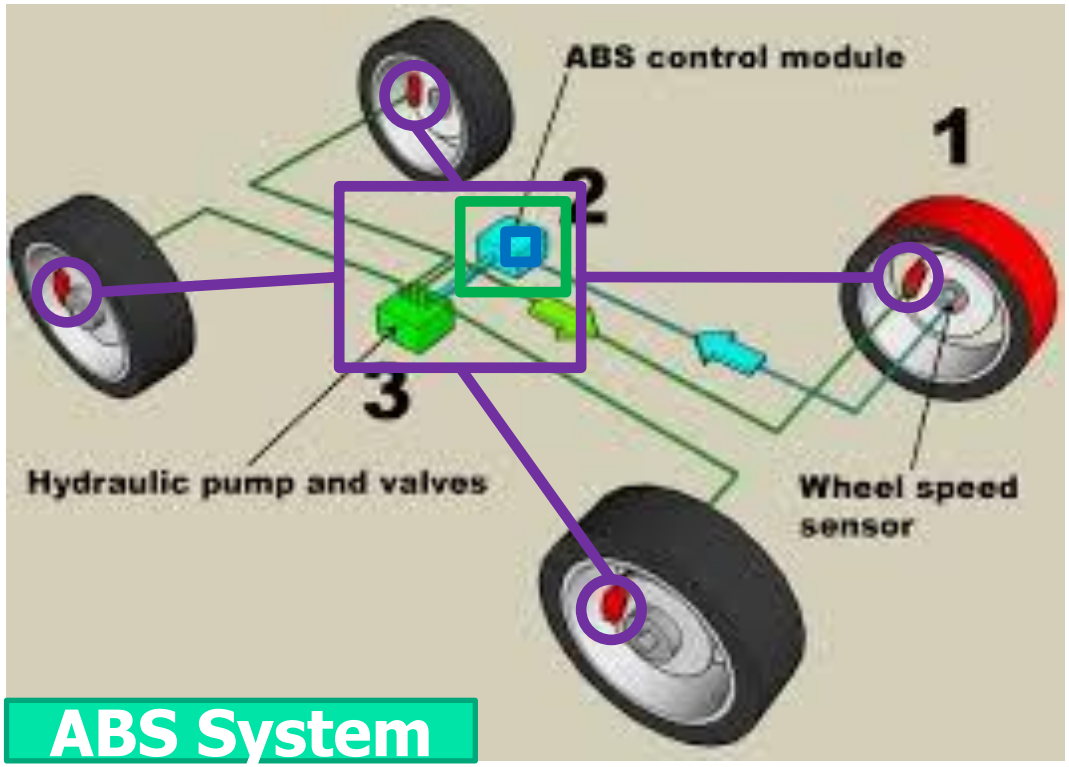
Systems Engineering Analysis

Functional Importance

1st Control System

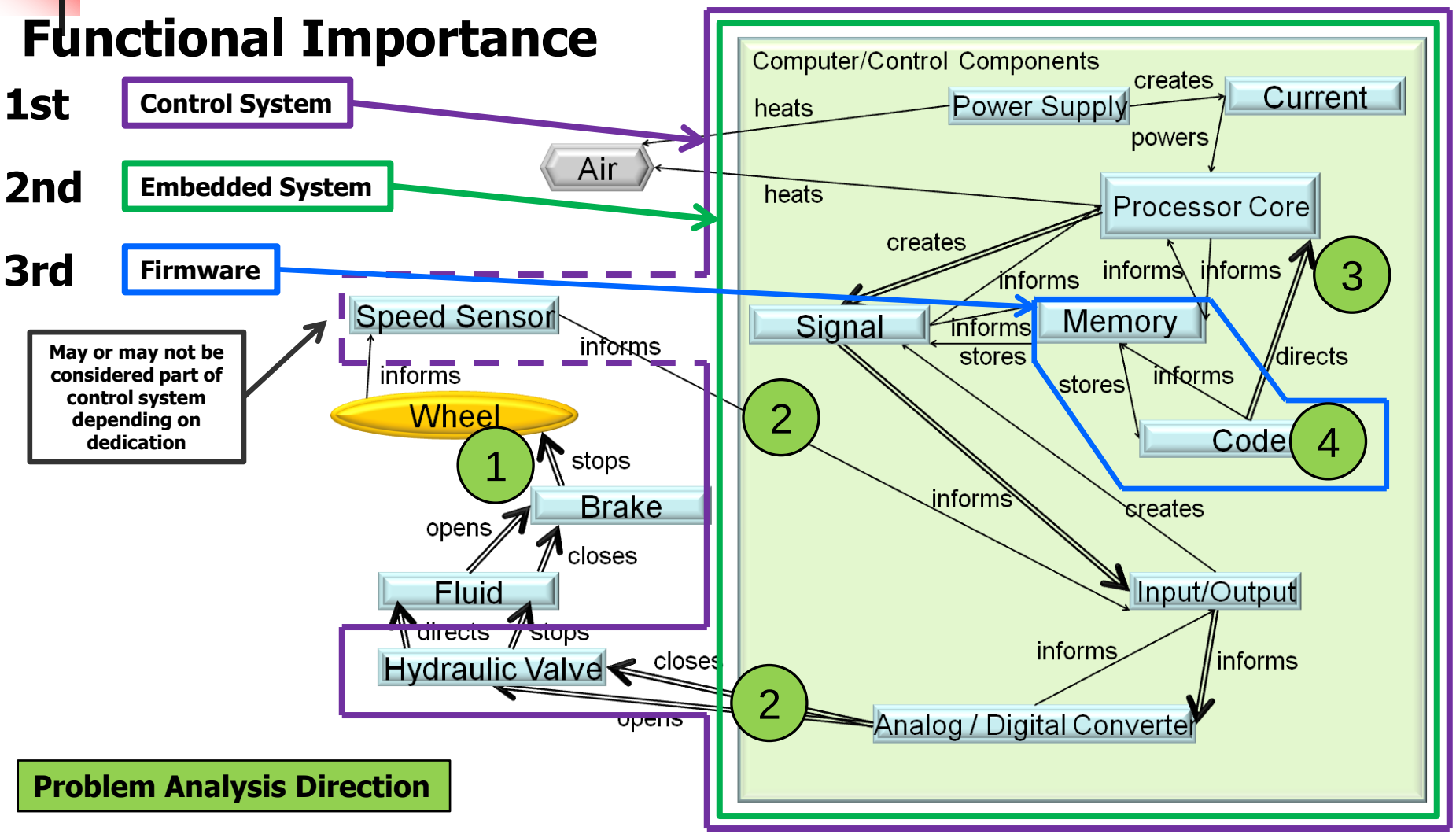
2nd Embedded System

3rd Firmware



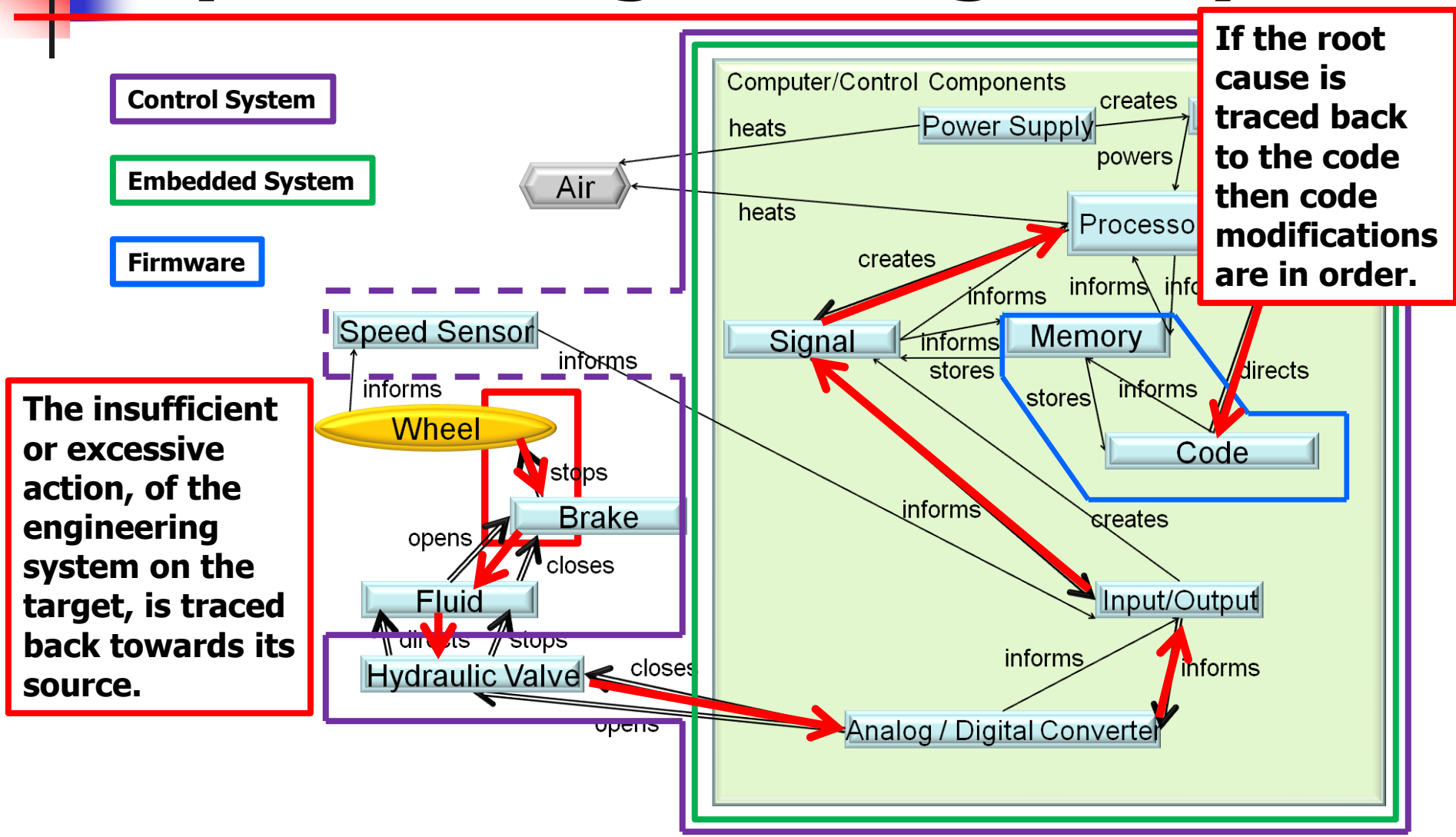
Concentric Hierarchy of Control System

Systems Engineering Analysis



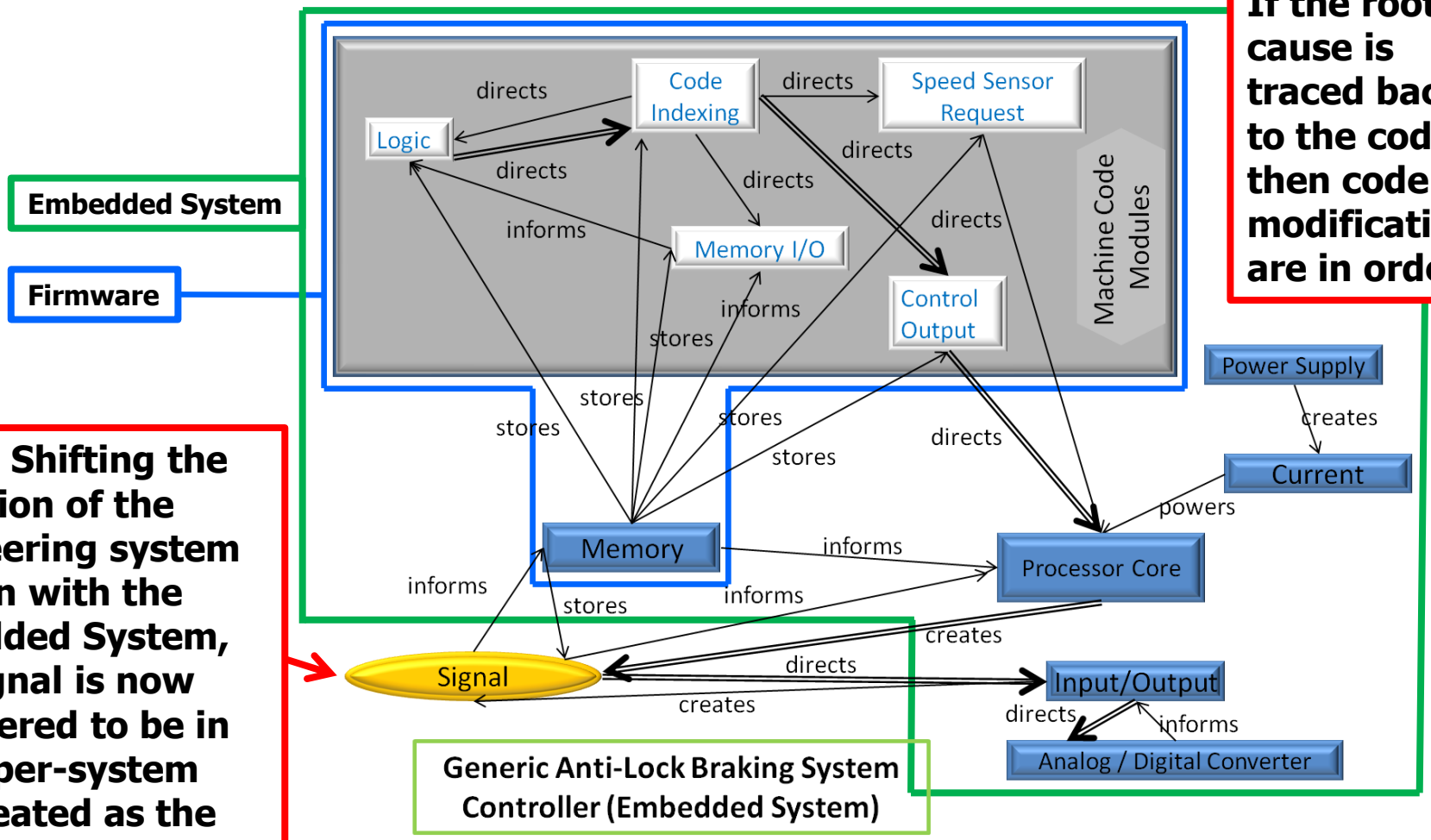
Concentric Hierarchy of Control System / Analysis Direction

Systems Engineering Analysis



Concentric Hierarchy of Control System / Analysis Direction

Systems Engineering Analysis

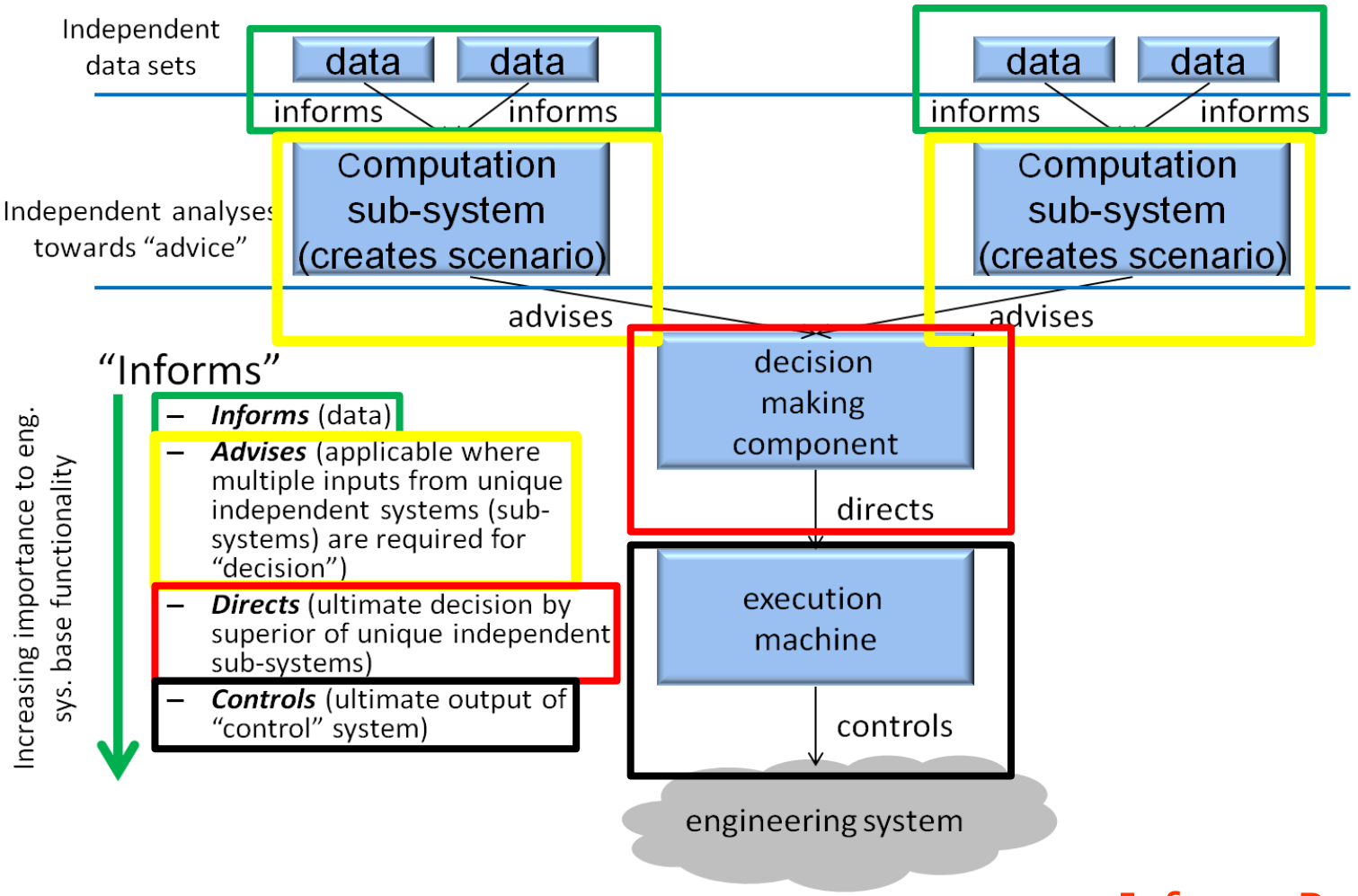


If the root cause is traced back to the code then code modifications are in order.

Note – Shifting the definition of the engineering system to align with the Embedded System, the Signal is now considered to be in the super-system and treated as the target.

Coding Modules

Additional Techniques



Informs Resolution

Additional Techniques

SI for Engineering		SI for Computing		
Technical		Software/Firmware		Hardware
(MATChEM +)	*Abstraction**	*(SID-LC)*	*Abstraction*	
Mechanical	Object	Software/Firmware Package - design	Design	Operational System
Acoustic	Surface	Interface/Handshaking (w/external systems)	Protocol	Connections/Interconnects/Transmission
Thermal	Lattice/Matter	Code Dynamics (movement/action relative to "itself")	Execution	Clock/Speed/Orchestration
Chemical	Molecular	Language/OS	Instruction	Component/Appliance (micro and macro)
Electro-Magnetic	Electron/Spin	Coding (routines/objects/agents)	Objects	Transistors/Devices
		Bits/Words	Object Code	Electrons/Holes
+				
Nuclear	Atomic			
Biological	*various*	n/a	n/a	Machine/Human Interface

* The Techncial Abstraction column is a modifciation of work by Gregory Frenklach

Resource Analysis

Summary

- Functionally model the engineering system the computing sub-system is associated with
- Understand and treat the control system, embedded system and the coding as separate entities
- Address the "issue" at the root (i.e., only modify the coding if that is indeed the problem)
- Utilize the delineation of: informs, advises, directs and controls
- Look for system resources to support solution generation (analogous to MATChEM)

References

- [1] Conley, David (2013) SI for Computing Advanced Course - C4 Rev 6. *Innomation Corp training material.* Slide 22
- [2] Conley, David (2013) SI for Computing Advanced Course - C4 Rev 6. *Innomation Corp training material.* Slide 36
- [3] Conley, David (2013) SI for Computing Advanced Course - C4 Rev 6. *Innomation Corp training material.* Slide 37
- [4] Conley, David (2013) SI for Computing Advanced Course - C4 Rev 6. *Innomation Corp training material.* Slides 76-80 & 121
- [5] Conley, David (2013) SI for Computing Advanced Course - C4 Rev 6. *Innomation Corp training material.* Slide 85 & 121
- [6] Conley, David (2013) SI for Computing Advanced Course - C4 Rev 6. *Innomation Corp training material.* Slide 194
- [7] Conley, David (2013) SI for Computing Advanced Course - C4 Rev 6. *Innomation Corp training material.* Slide 100
- [8] Conley, David (2014) SI for Computing Advanced Course - C4 Rev 7. *Innomation Corp training material.* Slide 450

Contact Info

David W. Conley
David@innomationcorp.com
+01-505-206-3401

INNOMATION CORP
Systematic Technology & Business Innovation

David@TRIZPQRGroup.com