AMS

A very short introduction to

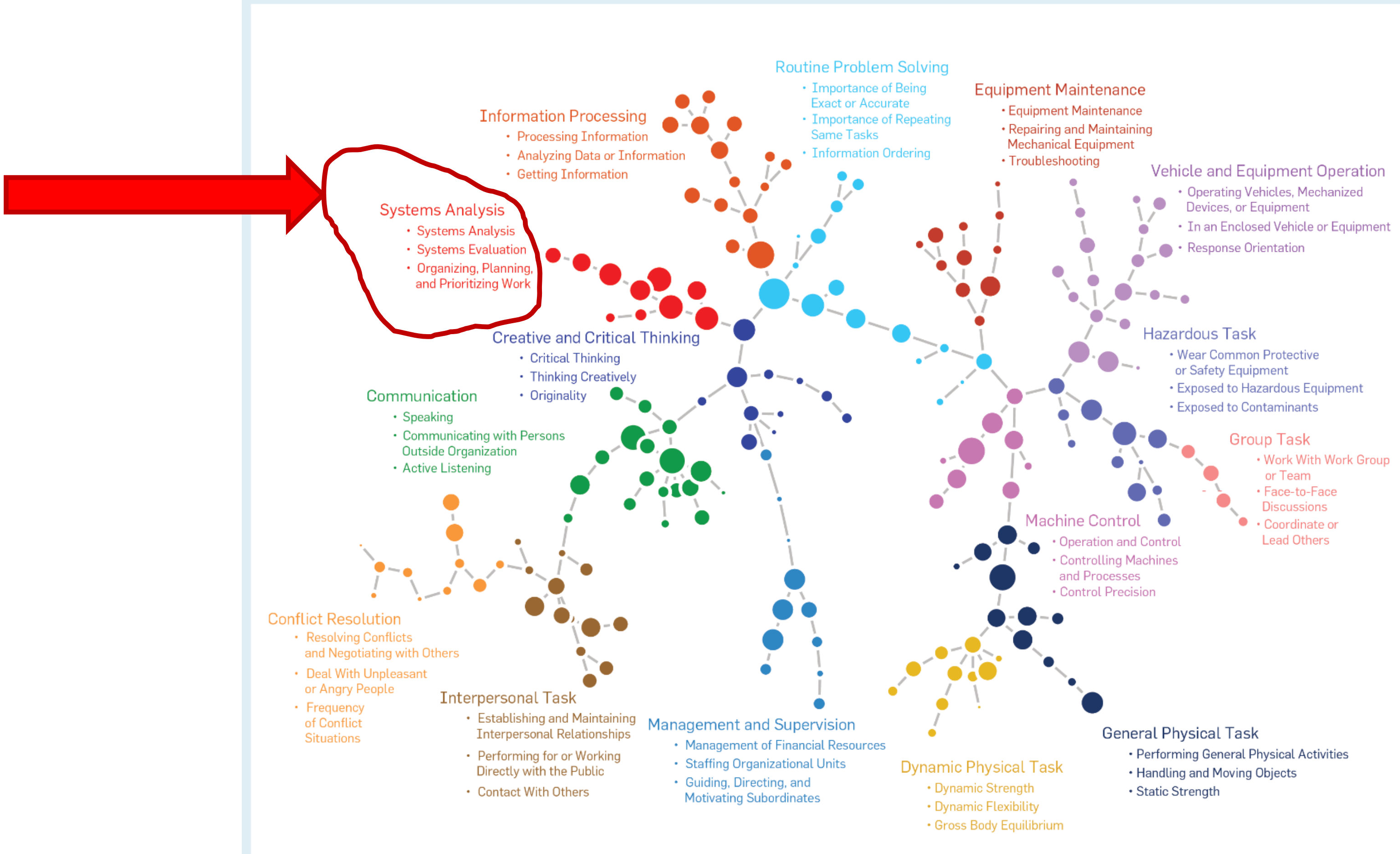
Systems Engineering

&

Conceptual Modelling

AMS in the data-driven exploration!

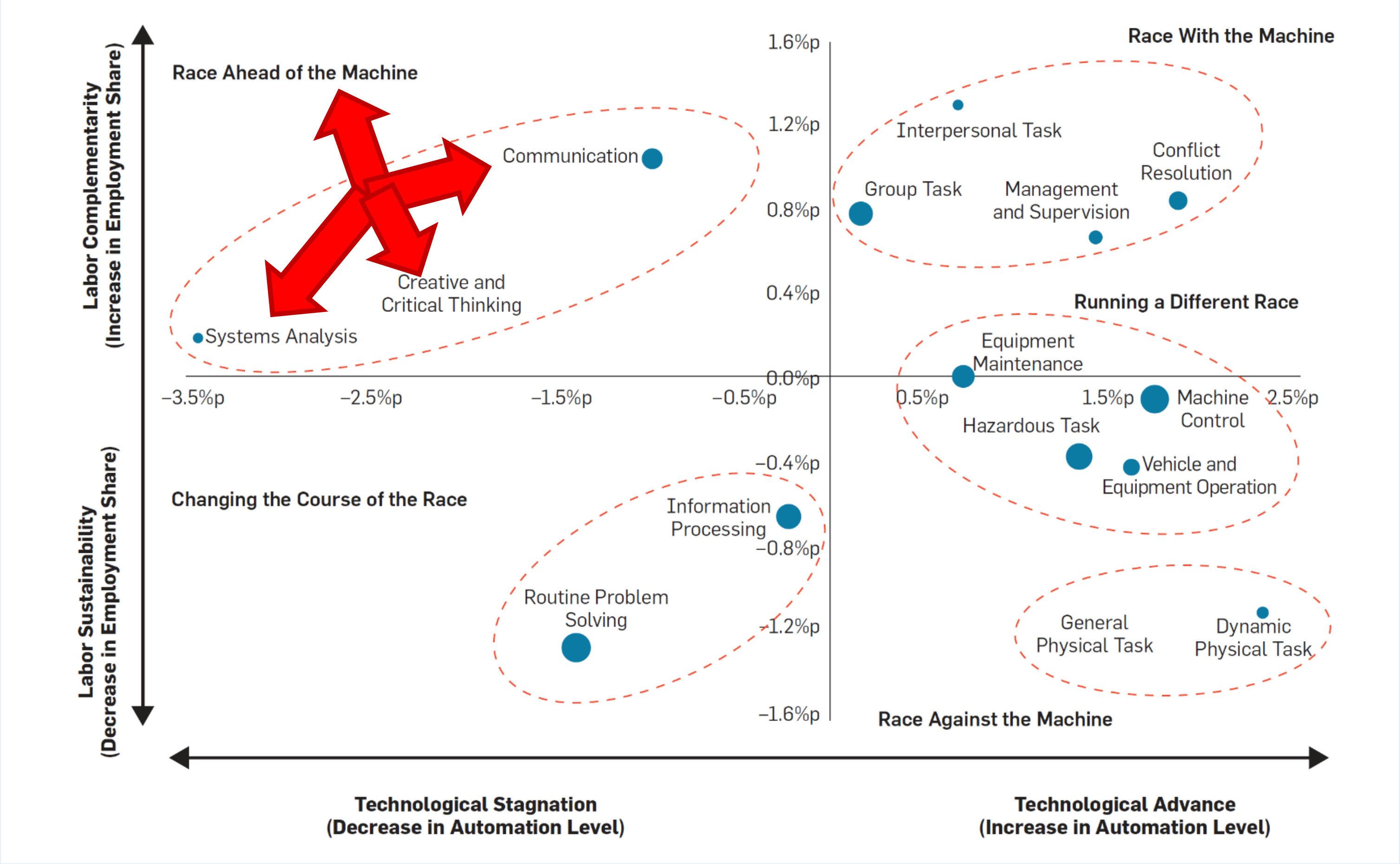
The graph shows the minimum spanning tree of a skill network constructed using 2020 occupational data; a task type is equivalent to a cluster of skills. The nodes are colored according to task types (clusters). Each cluster is named based on skill components. Node size is proportional to each skill's average importance across occupations. The full list of skill components for each task type is presented in the online appendix. The interactive skill network can be found at www.jobautomationindex.com.



Park, J., & Kim, J. (2022). A data-driven exploration of the race between human labor and machines in the 21st century. *Communications of the ACM*, *65*(5), 79-87.

AMS in the data-driven exploration!

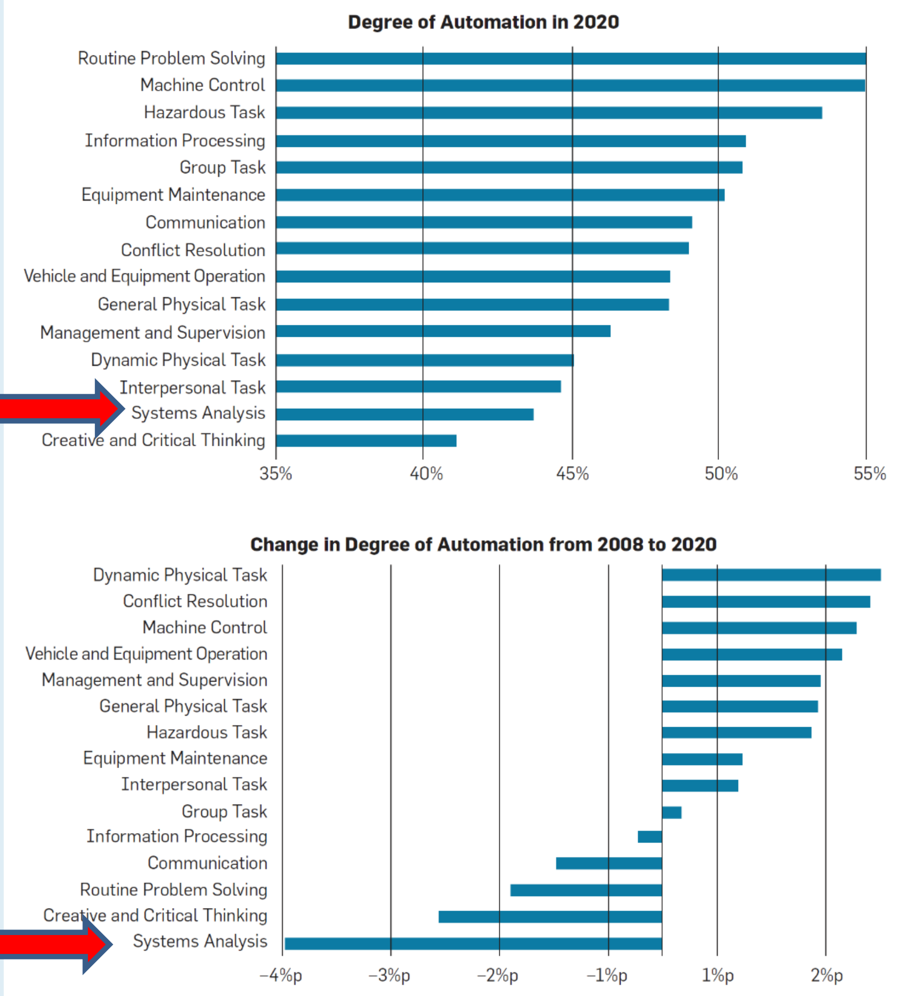
This taxonomy of task automation is presented from the viewpoint of the race between human labor and machines. The *x*-axis indicates the change in the degree of automation of each task from 2008 to 2020: the *y*-axis indicates the average within-industry changes in employment share for each task from 2008 to 2020. Point size is proportional to the degree of automation in 2020.



Park, J., & Kim, J. (2022). A data-driven exploration of the race between human labor and machines in the 21st century. *Communications of the ACM*, *65*(5), 79-87.

AMS in the data-driven exploration!

Figure 3. Task-level degree of automation.

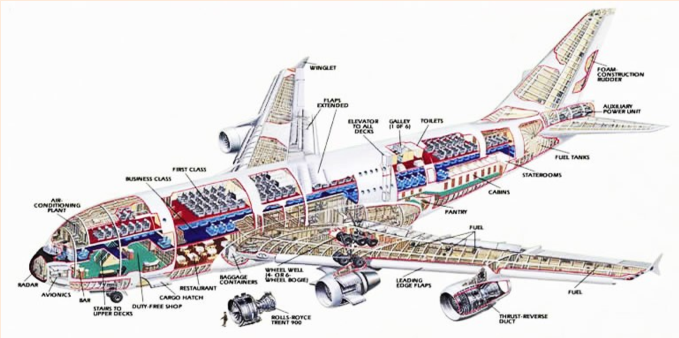


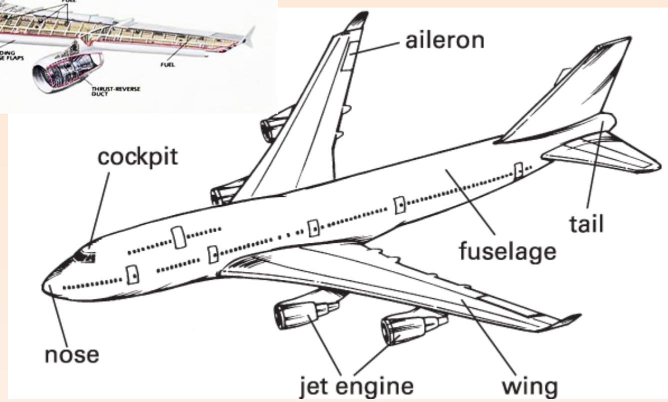
Task with a low level of automation...???...

Meaning: YES, IT REQUIRES HUMAN EFFORT !!! !!! !!! !!!

Discuss in class...

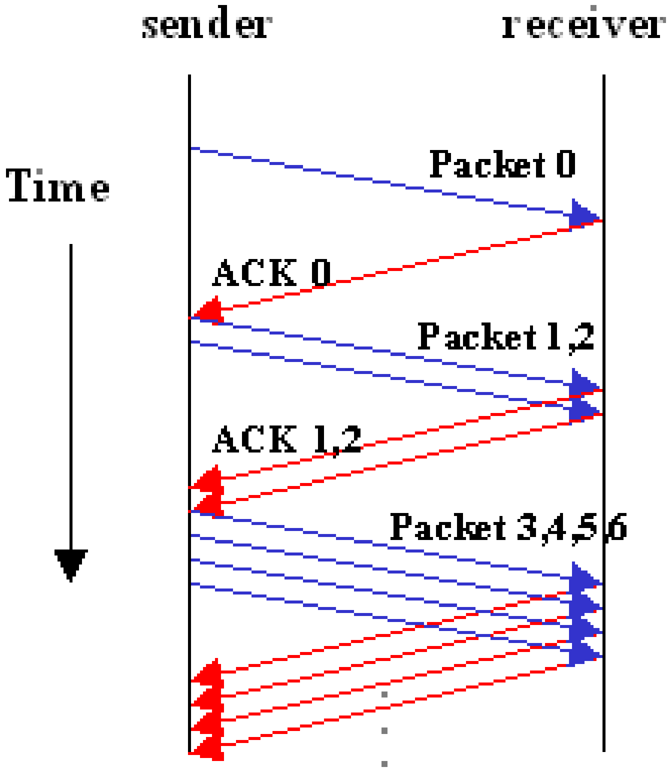
Park, J., & Kim, J. (2022). A data-driven exploration of the race between human labor and machines in the 21st century. *Communications of the ACM, 65*(5), 79-87.





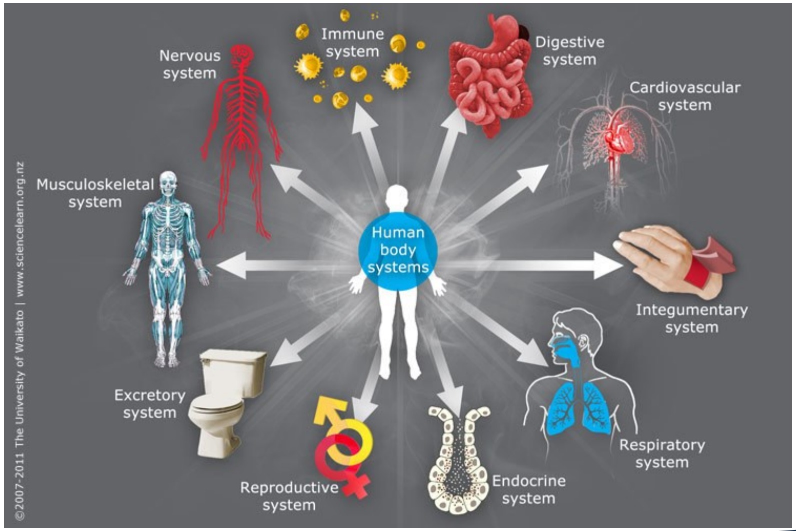
AMS

A very short address of Systems Thinking and Systems Engineering …



to start... what are we going to address in AMS?

What does it really mean if someone says a “thing” is a “system”? Concept to retain: “concern”!!! Discussion...

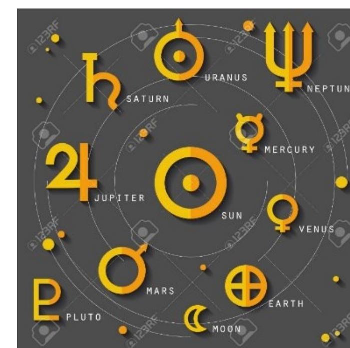


…what does it really mean if someone says “the human body is made of a set of systems” ???…

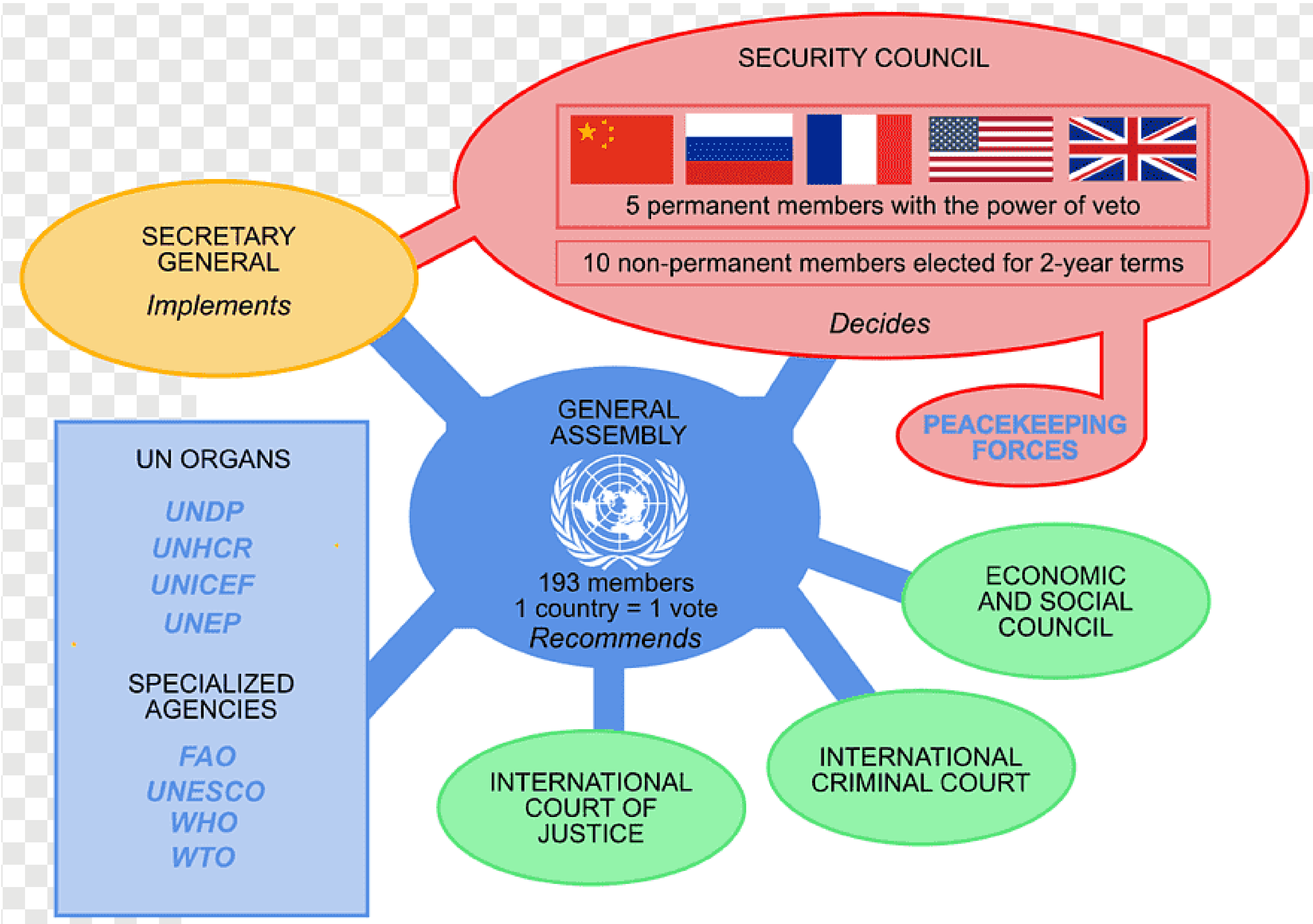
...what does it really mean if someone says “the **solar system** is a system”?

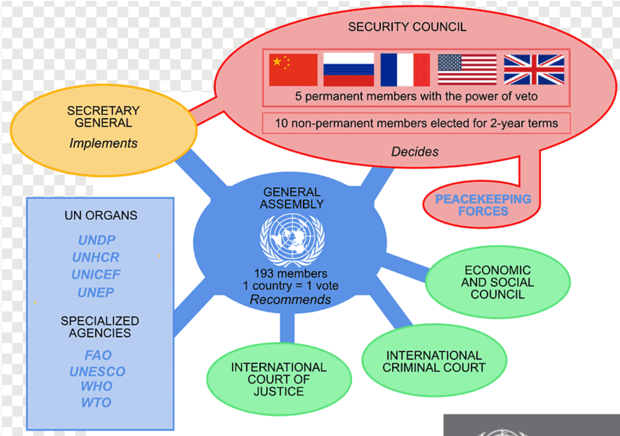
Are these both “looking at” the same “thing”?

Do both have the same “concerns”?

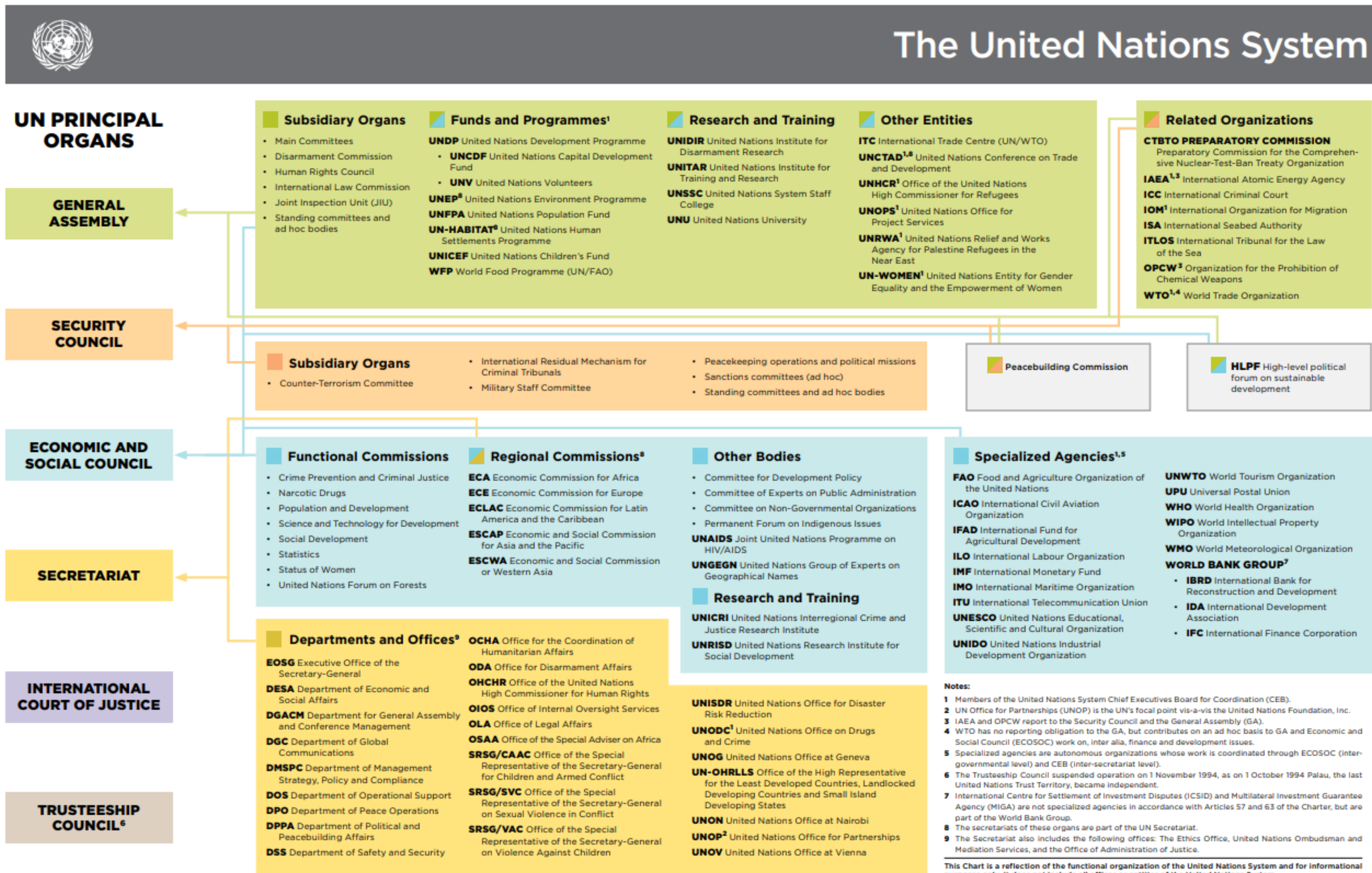


An example of an "organizational system", the "UN System"
(btw, here we only see the "structural view" of this system… nothing is shown about its "dynamic view"…)



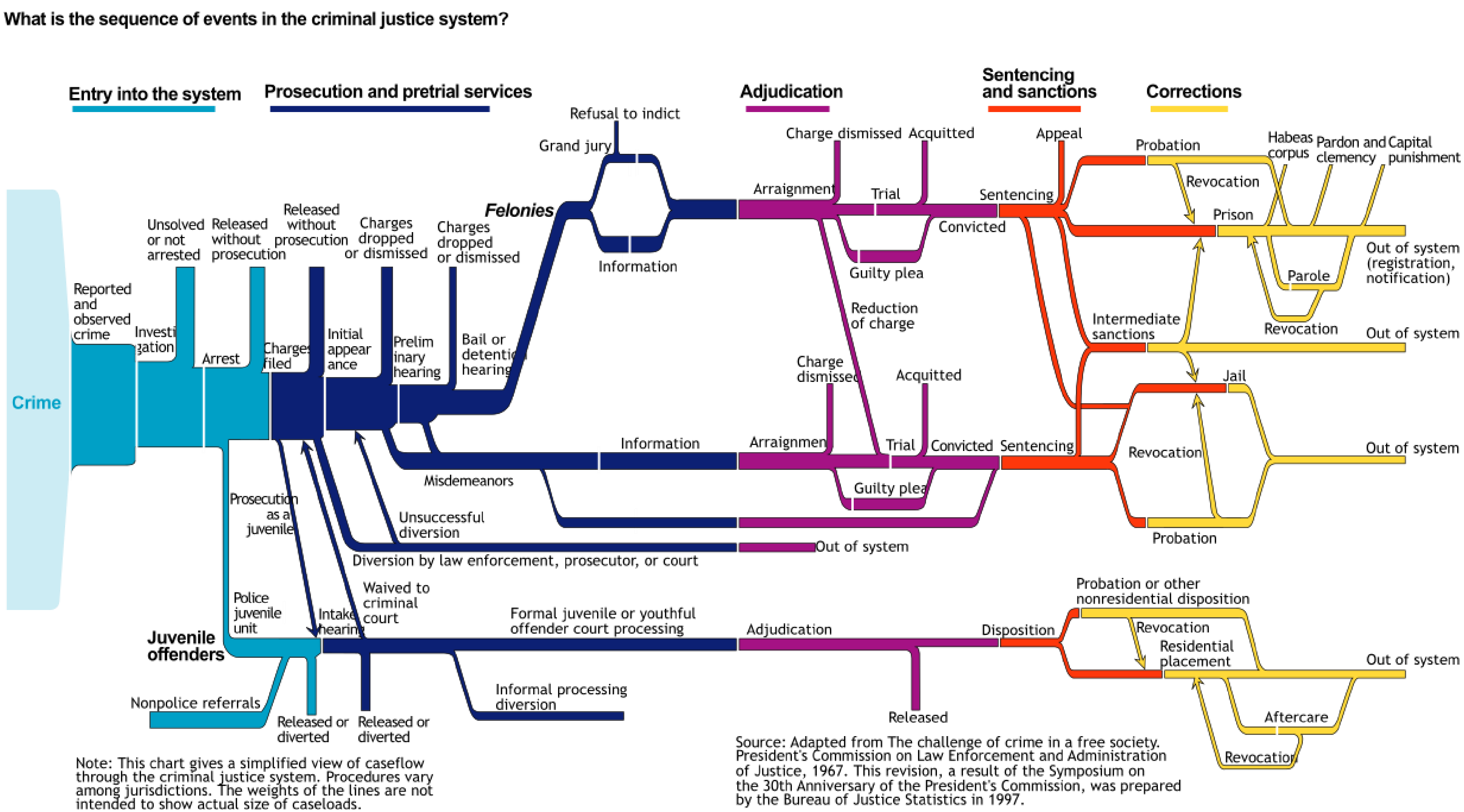


One other view the "UN System"
(still, a "structural view"…)
What is the best?
Can we claim one is better than the other?
Let us discuss that… ;-)



BTW, for curiosity, explore more at: https://www.un.org/en/about-us/un-system

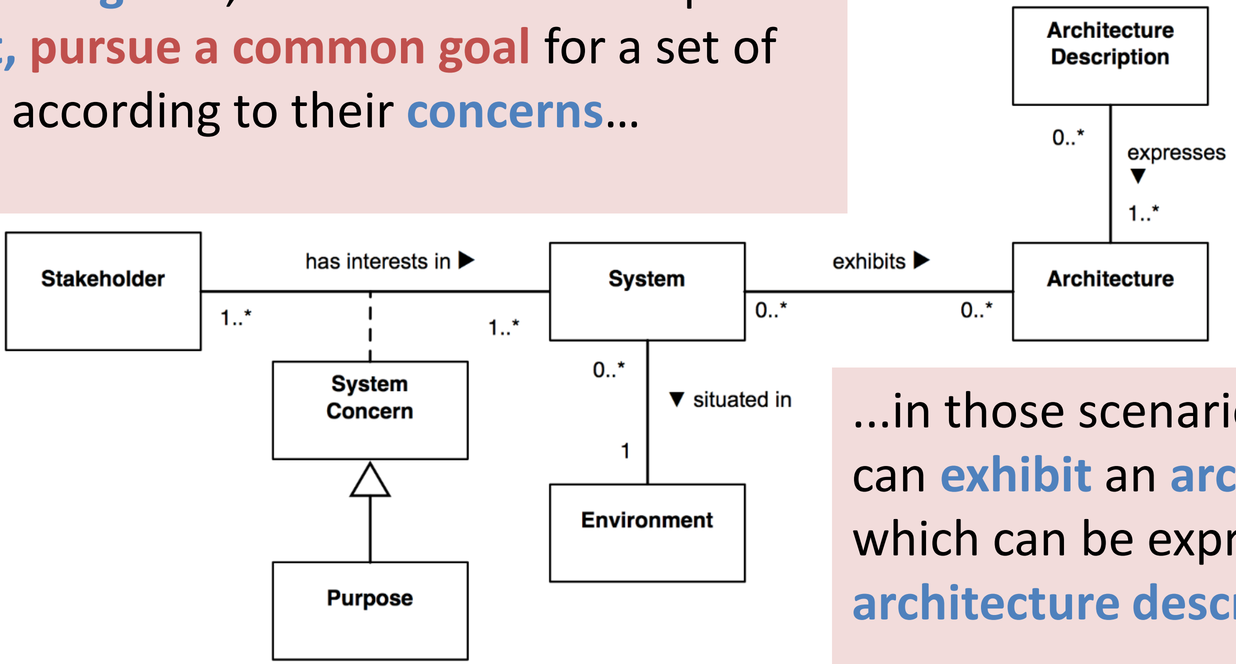
Now, a “dynamic view” of a system, the “US criminal justice system” (a flowchart…)



https://en.wikipedia.org/wiki/Criminal_justice#/media/File:Cjsflowco.svg

About the concept of “system”

A system is a thing that, when situated in a specific environment, pursue a common goal for a set of stakeholders according to their concerns…



…in those scenarios, a system can exhibit an architecture, which can be expressed in an architecture description…

Important: a “system” is not something that exists “per si”, but something that “emerges conceptually” from the process of giving attention to concerns of stakeholders…

Explore:

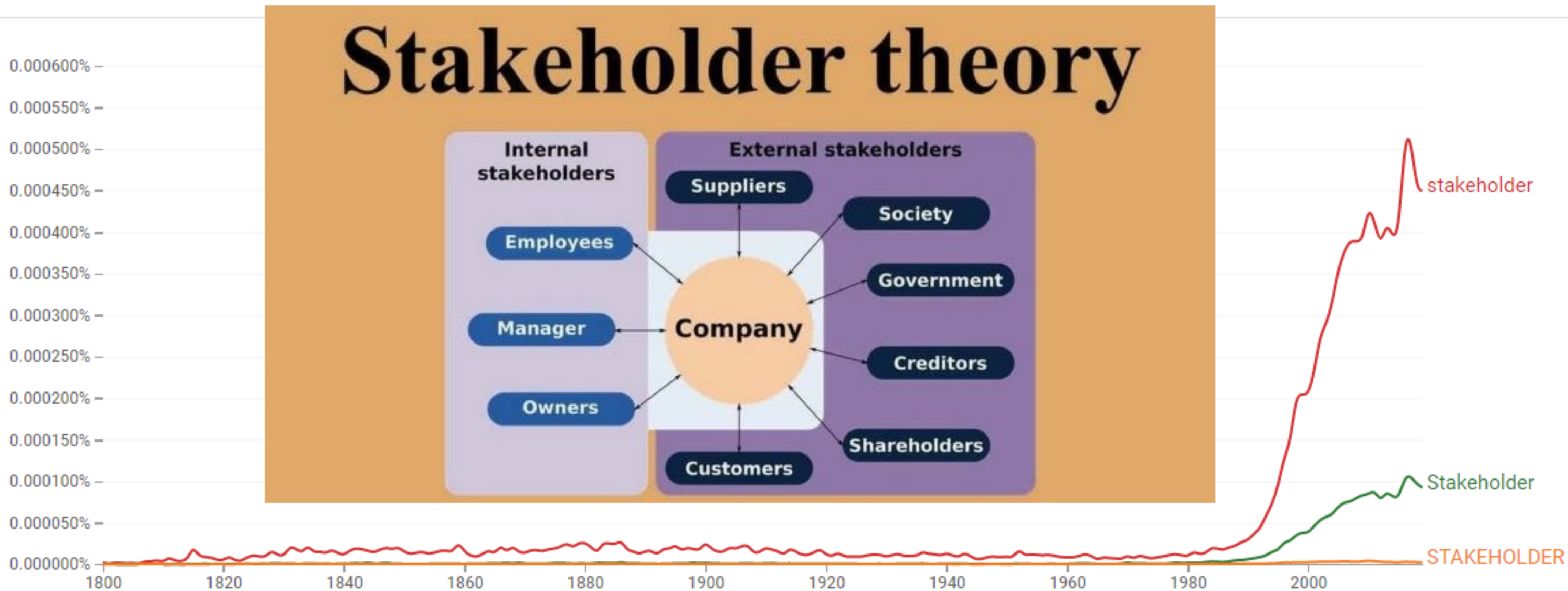
- *ISO/IEC/IEEE 42010, Systems and software engineering — Architecture description*
- *https://www.iso-architecture.org/*

...thus, it all depends of "who cares" => the mighty STAKEHOLDER !!!

Google Books Ngram Viewer

stakeholder

1800 - 2019 | English (2019) | Case-Insensitive | Smoothing of 0



(click on line/label for focus, right click to expand/contract wildcards)

Search in Google Books

stakeholder > 1800 - 1883 | 1884 - 2012 | 2013 - 2015 | 2016 - 2017 | 2018 - 2019 | English (2019)

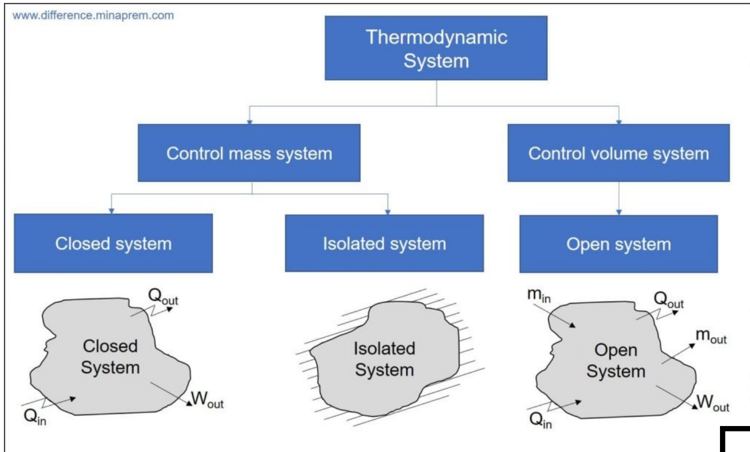
More about "system"

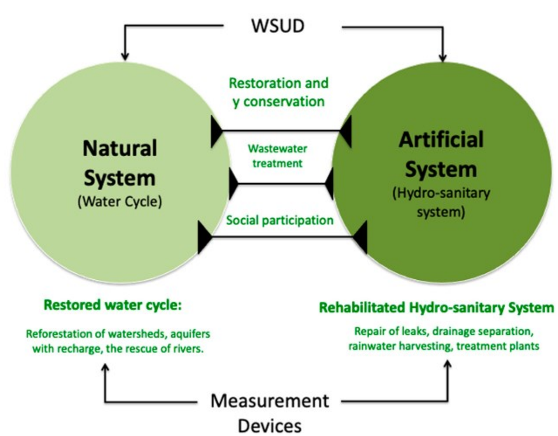
In this course we'll understand a system as a **thing** made of a **set of entities** that can be software, hardware, people, business procedures, or any combination thereof…

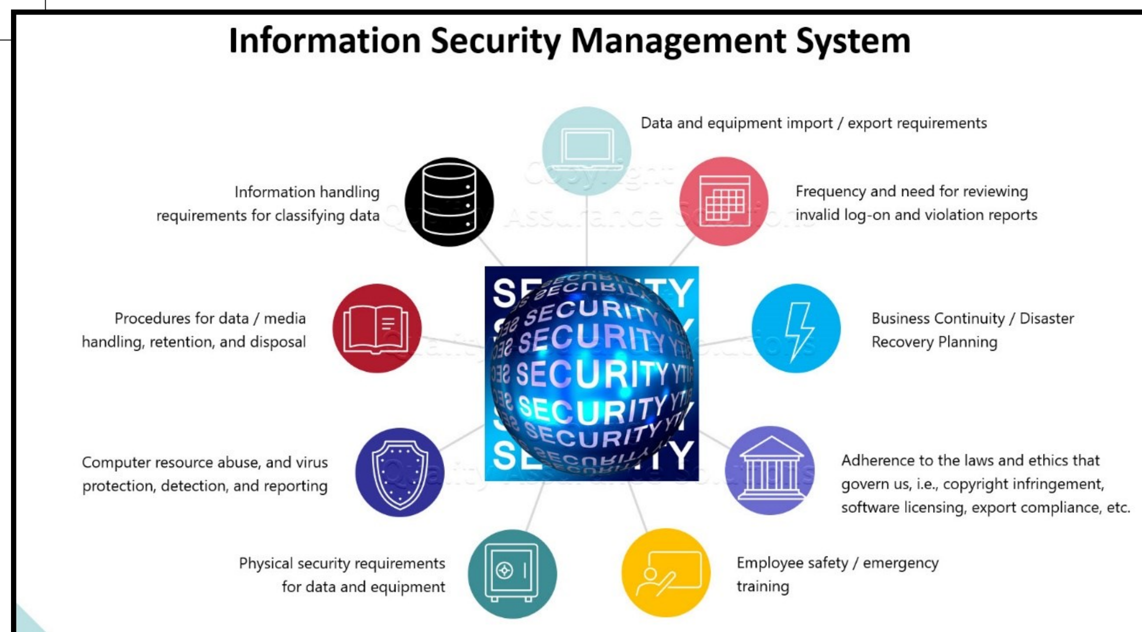
In other words, we'll care about systems that are:

- "socio-technical", meaning they are made of People, Processes and Technology
- "complicated", meaning they are made of a good number of different entities

“systems” everywhere…

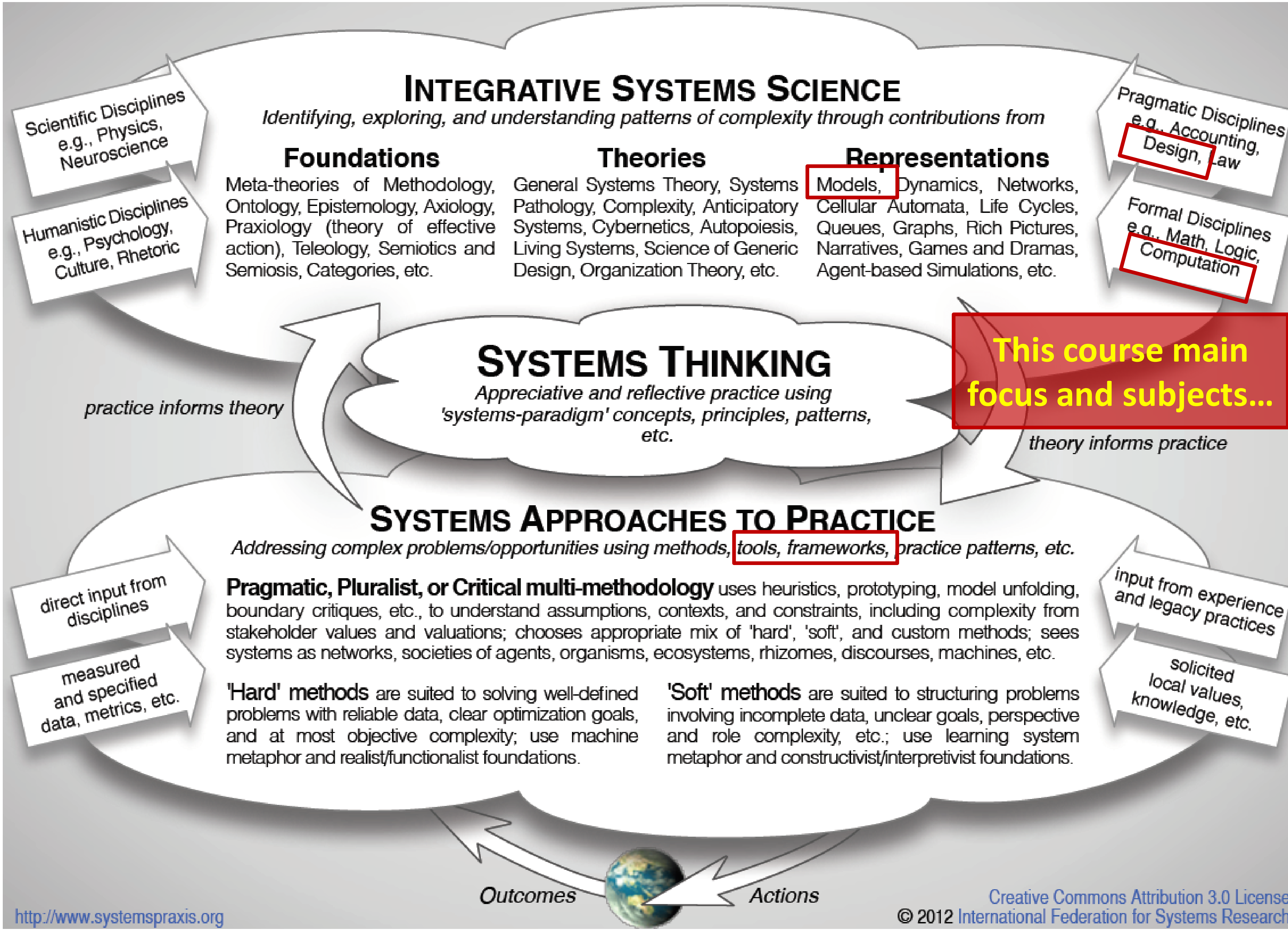




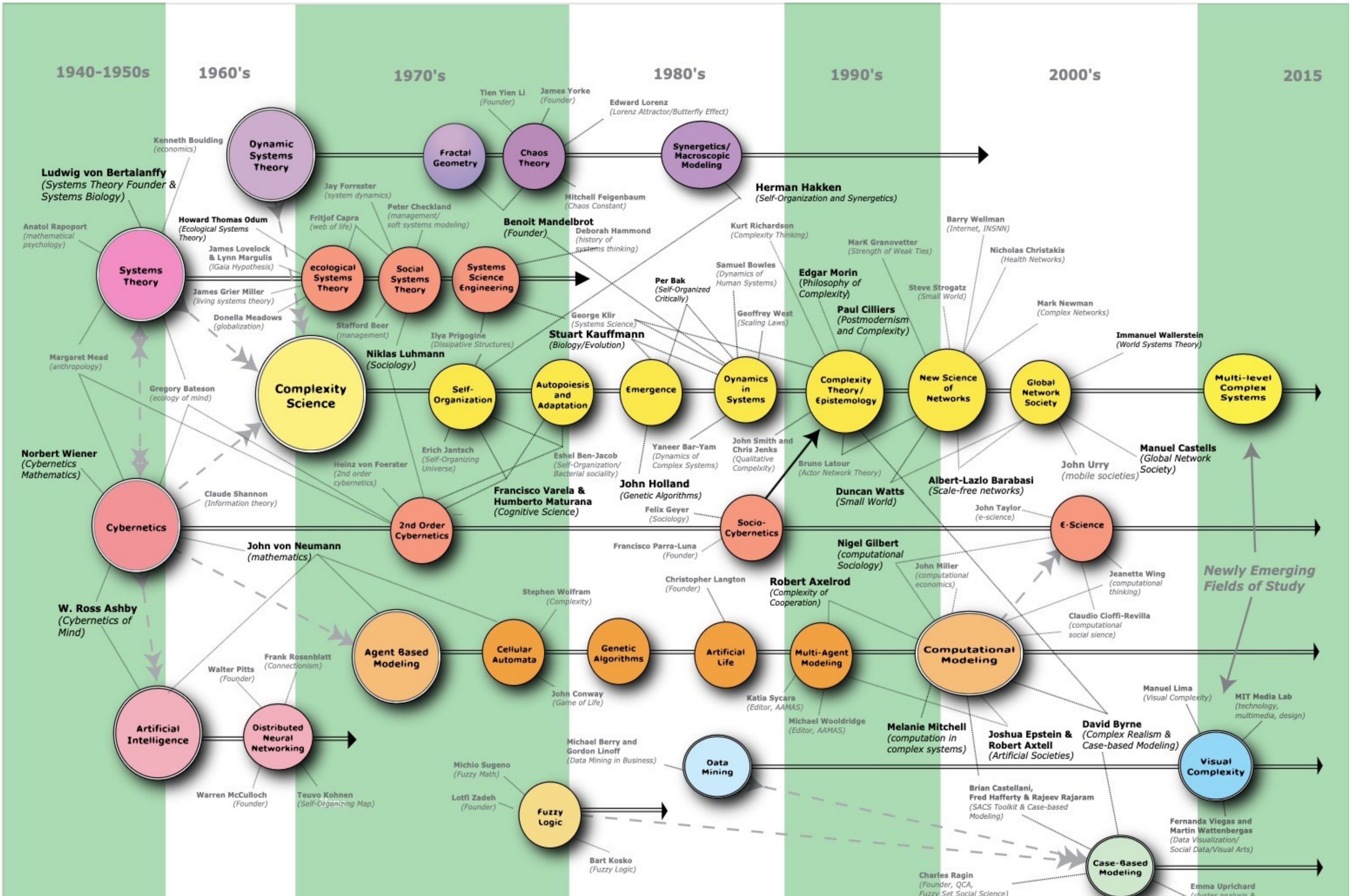


http://www.difference.minaprem.com/thermodynamics/difference-between-open-system-closed-system-and-isolated-system/
https://www.mdpi.com/2073-4441/13/5/601/htm
https://fujisoft.com/going-for-cctv-security-system-for-your-business-here-are-some-valuable-tips/
https://www.quality-assurance-solutions.com/information-security-management-system.html

Extra: (http://systemspraxis.org) The Systems Praxis Framework



One historical view of “system thinking”…



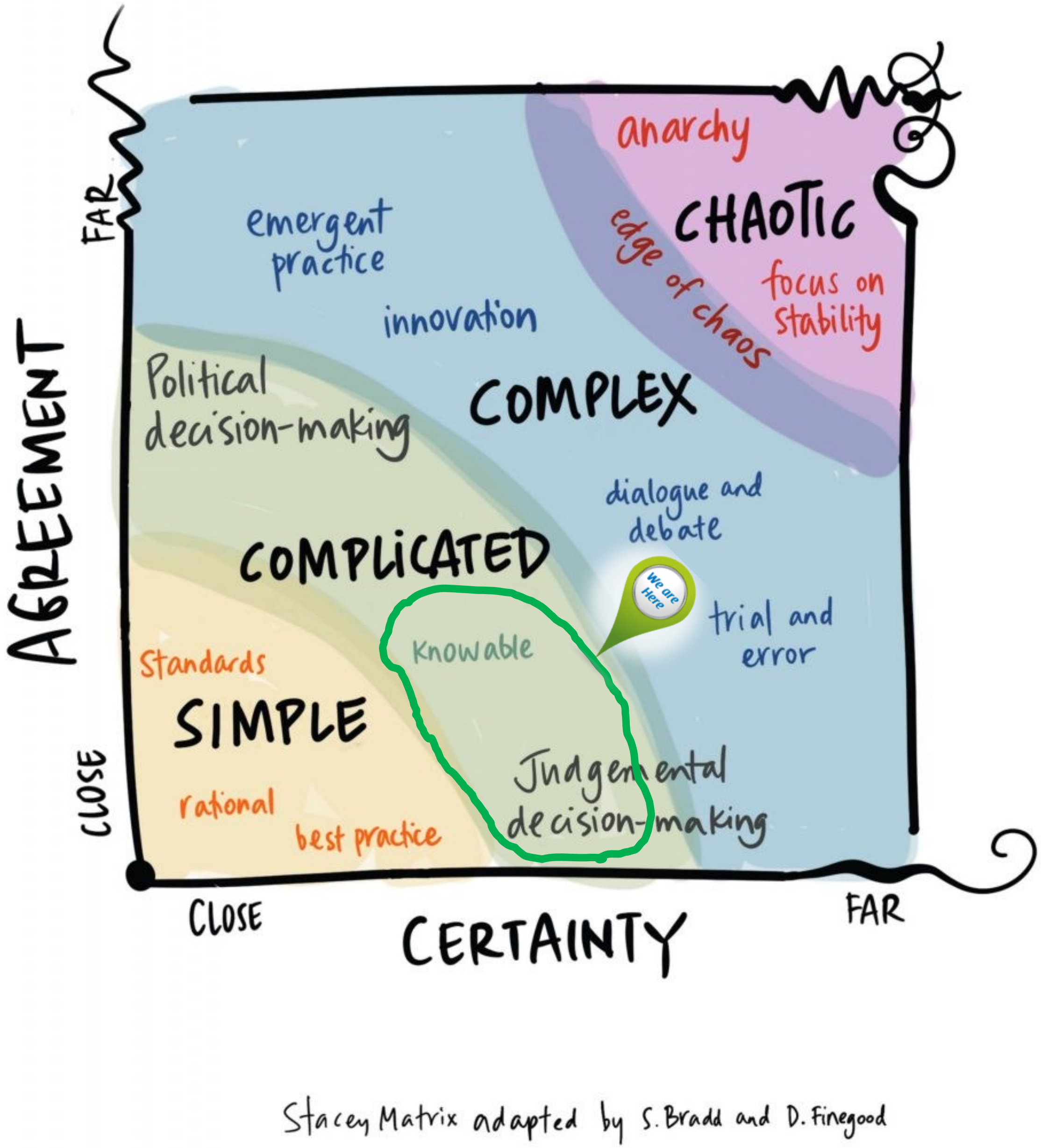
https://chaturvedimayank.wordpress.com/tag/eugene-stanley/

One more way to classify “systems” from a “systems engineering” perspective

Hard systems	Soft systems
1. Well-defined problem. 2. Stakeholders share a common view of the problems.	1. Problem is loosely defined due to an un-objective reality. 2. Stakeholders interpret problems differently.
3. Scientific approach to problem-solving.	3. Several approaches to problem-solving including scientific approaches.
4. Mostly deals with engineering-related factors.	4. Socio-technical and human factors also need to be considered.
5. The usual purpose is finding an ideal solution that addresses all the concerns of the stakeholders.	5. Finding an ideal solution may not be goal. 6. Alternative purposes include learning and better understanding the system.

“Simple, Complicated and Complex Decision-Making”

https://drawingchange.com/project/simple-complicated-and-complex-decision-making-new-visual/



(...)

Certainty and agreement

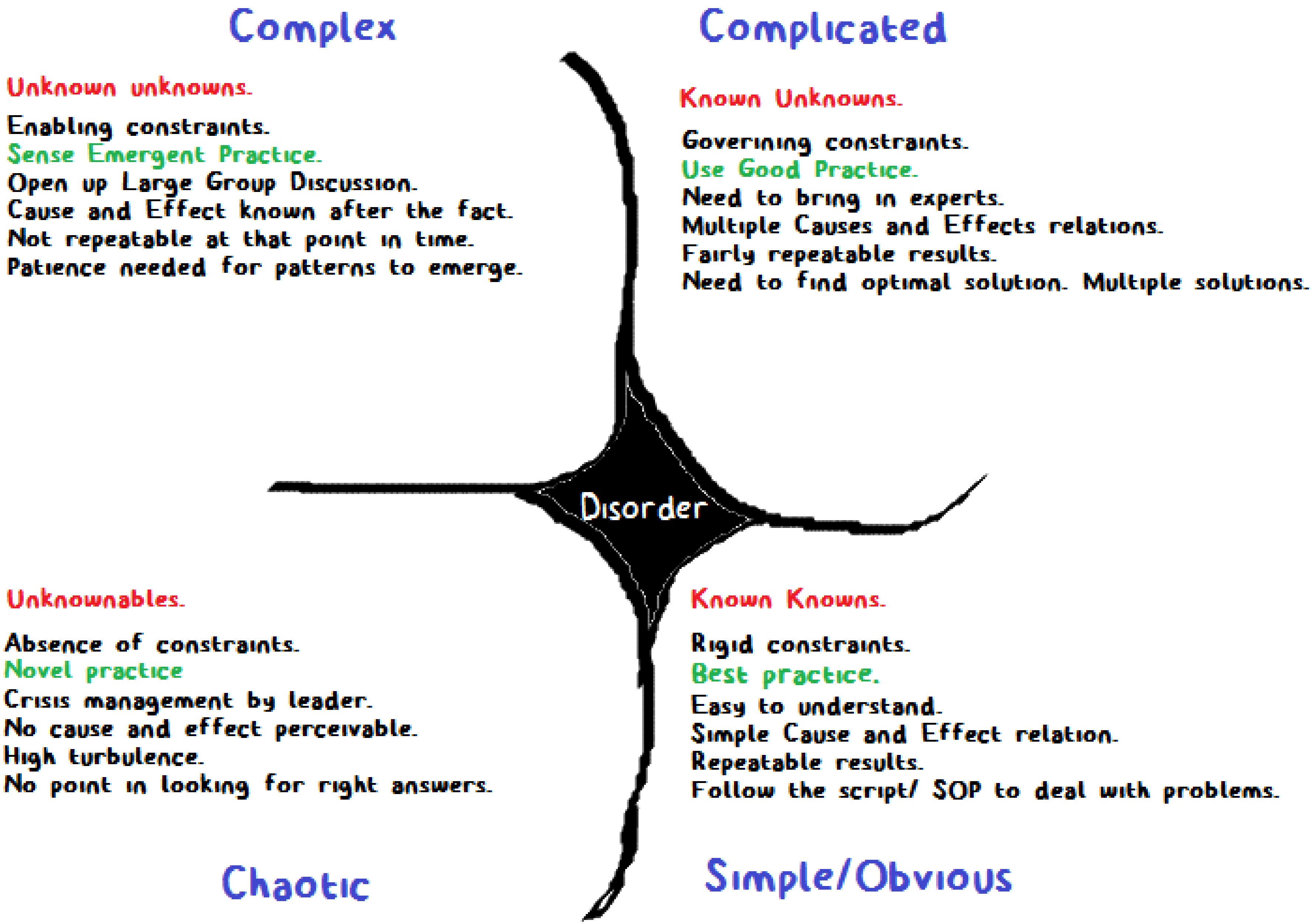
Sometimes we try and treat organizations — or worse, people! — as if they are problems to be solved. We pretend that organizations are “simple” or that there’s one way to solve a big issue like homelessness. And, the further we get from agreement and certainty … the more chaos can also emerge. How can we facilitate through this? Why does some information always seem to be changing?

Visuals for Complexity

It could be that we need decisions with complex problems – not for simple problems. And the qualities that go along with complex decisions are different. Dr Diane Finegood teamed up with Drawing Change to update this Stacey Matrix visual about simple, complicated and complex decision making, featuring visual embellishments and new colours to highlight the stages.

(...)

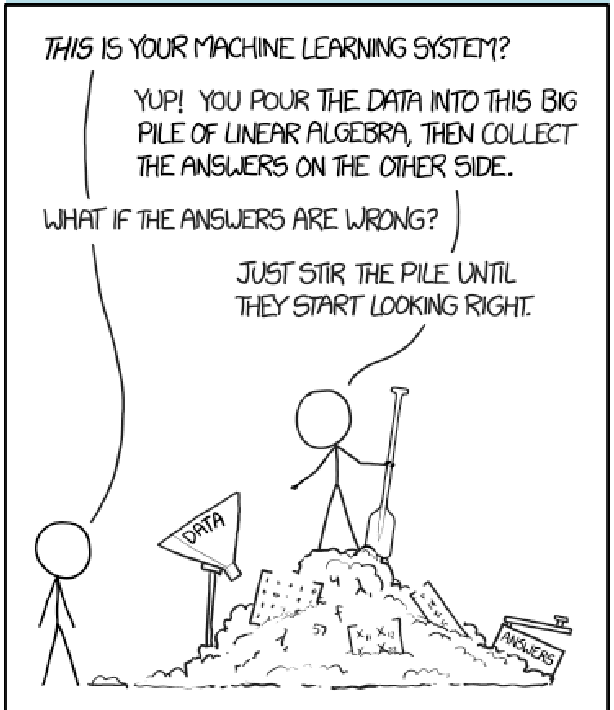
Keep calm… making it simple with the **Cynefin framework…**



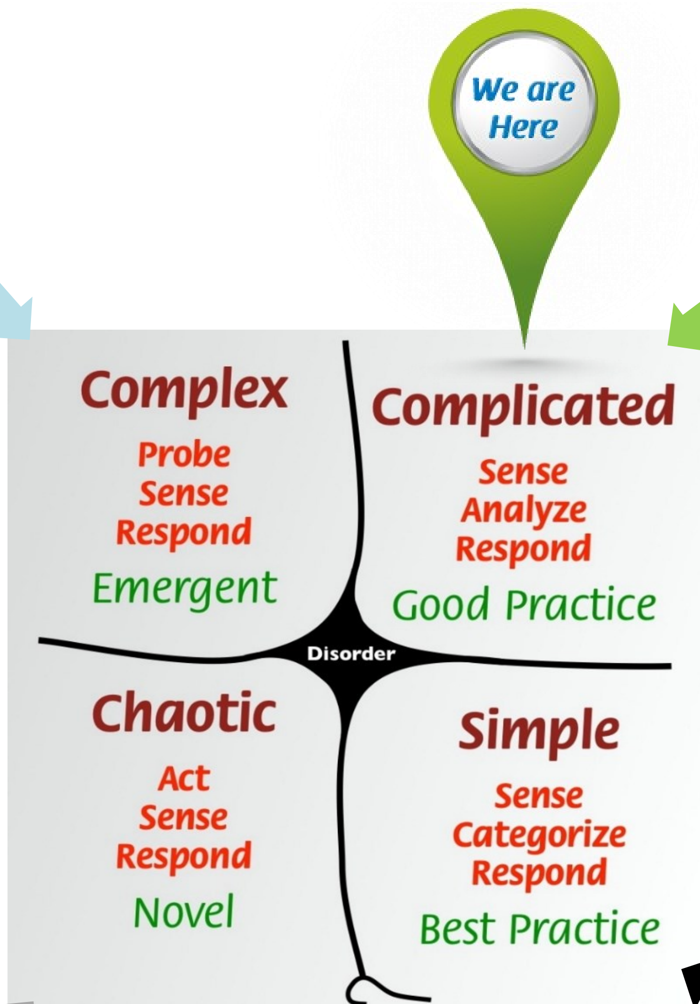
https://en.wikipedia.org/wiki/Cynefin_framework
https://harishsnotebook.files.wordpress.com/2015/09/cynefin-final.png

The **Cynefin framework...**

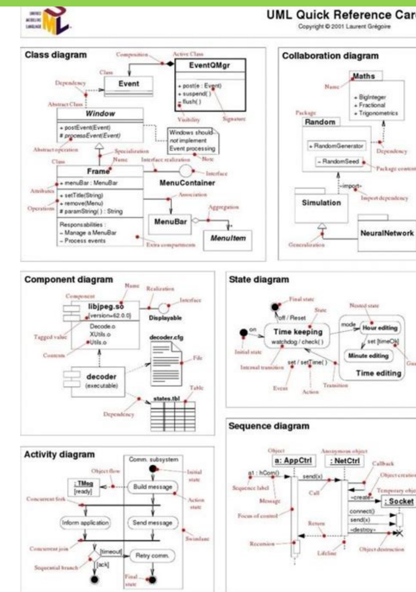
…"cause and effect" can only be perceived in retrospective (we ca guess rules for change state…)



…not possible to understand "cause and effect" (it can jump from any state to any state, and we have no idea of why and how…)



…"cause and effect" can be assured or discovered with expert knowledge (we can enforce the rules change state…)



…"cause and effect" is obvious (the "state machine" is easily and clearly perceived…)

https://en.wikipedia.org/wiki/Cynefin_framework
https://www.mindtools.com/pages/article/cynefin-framework.htm
http://varianceexplained.org/r/ds-ml-ai/

More definitions of System…

- **ISO/IEC 15288:2008**: "A combination of interacting elements organized to achieve one or more stated purposes."
- **ANSI/EIA-632-1999:** "An aggregation of end products and enabling products to achieve a given purpose."
- **IEEE 1220-1998:** "A set or arrangement of elements and processes that are related and whose behavior satisfies customer/operational needs and provides for life cycle sustainment of the products."
- **NASA Systems Engineering Handbook:** "(1) The combination of elements that function together to produce the capability to meet a need. The elements include all hardware, software, equipment, facilities, personnel, processes, and procedures needed for this purpose. (2) The end product (which performs operational functions) and enabling products (which provide life-cycle support services to the operational end products) that make up a system."
- **INCOSE:** "A system is a construct or collection of different elements that together produce results not obtainable by the elements alone. The elements, or parts, can include people, hardware, software, facilities, policies, and documents; that is, all things required to produce systems-level results. The results include system level qualities, properties, characteristics, functions, behavior and performance. The value added by the system as a whole, beyond that contributed independently by the parts, is primarily created by the relationship among the parts; that is, how they are interconnected.“

(Weilkiens 2007)

Engineering Systems = Engineering … Systems

Engineering: *the work of **designing** and **creating** large structures (such as roads and bridges) or new products or systems by using scientific methods.*

***System**: group of **entities** that **interact** with a specific **purpose** that cannot be achieved by any of the entities alone.*

(Merriam Webster's Learner's Dictionary - http://www.learnersdictionary.com/search/engineering)

Systems Engineering

"Systems Engineering is an **interdisciplinary approach** and **means** to enable the **realization of successful systems**." (INCOSE, 2004)

"Systems engineering is a discipline that concentrates on the **design and application of the whole (system)** as distinct from its parts. It involves looking at a problem in its entirety, taking into account all the facets and all the variables relating the social to the technical aspect." (Federal Aviation Agency (USA FAA). Systems Engineering Manual, 2006)

Systems Engineering

“System engineering is a robust approach to the **design, creation, and operation of systems**. In simple terms, the approach consists of:

- **identification and quantification** of system goals,
- **creation** of alternative system design concepts,
- **selection** and **implementation** of the best design,
- **verification** that the design is properly built and integrated, and
- **post-implementation assessment** of how well the system meets (or met) the goals.”

(NASA Systems Engineering Handbook, 1995)

Some definitions

(ISO 42010:2011; IEEE 1471:2000 for historical relevance…)

- **architecting:** the activities of defining, documenting, maintaining, improving, and certifying proper implementation of an architecture.
- **architect:** the person, team, or organization responsible for systems architecture
- **architecture:** the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.
- **conceptual model:** a model representing concepts and relationships between them.
- **concern:** those interests which pertain to the system's development, its operation or any other aspects that are critical or otherwise important to one or more stakeholders.
- **mission:** a use or operation for which a system is intended by one or more stakeholders to meet some set of objectives.
- **model:** a purposely abstracted representation of a concern.
- **modelling:** the process of generating models.
- **modelling language:** an artificial language to express models, which is defined by a consistent set of rules (ideally clear, precise and unambiguous) to be used for interpretation of the meaning of the models.
- **notation:** language used to visualise a model.
- **stakeholder:** an individual, team, or organization (or classes thereof) with interests in, or concerns relative to, a system.
- **system:** a collection of components organized to accomplish a specific function or set of functions.
- **view:** a representation of a whole system from the perspective of a related set of concerns.
- **viewpoint:** a specification of the conventions for constructing and using a view. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis.

AMS

Systems Engineering…

according to the SEBok…

(please explore the references...)

(http://www.sebokwiki.org/) Systems Engineering

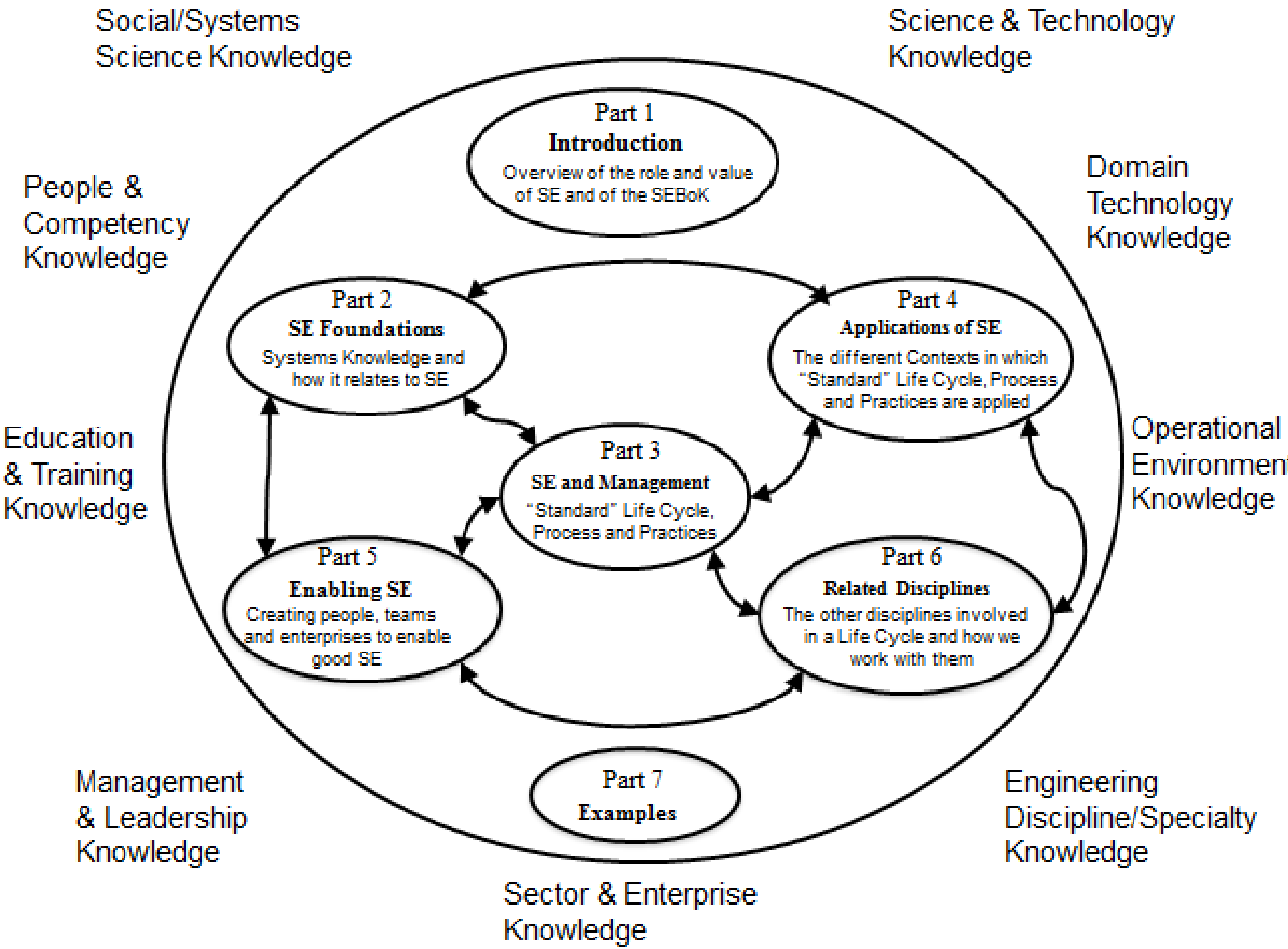
"Systems engineering is an interdisciplinary approach and means to enable the full life cycle of successful product, service and enterprise systems. It including problem discovery and formulation, solution definition and realization, and operational use, sustainment, and disposal. It can be applied to single problem situations or to the management of multiple interventions in commercial or public enterprises."

(..)

"Systems engineering has its roots in the fundamentals, principles, and models of foundational systems sciences, and associated management and engineering sciences. It is applied through the application of systems engineering processes within a managed life cycle working with a number of other management, engineering, and specialist disciplines. While traditionally applied to product development, systems engineering can also be applied to service and enterprise systems. As systems engineering is a collaborative approach, working with other engineering and management disciplines and specialisms, it relies on enabling competencies and structures at individual, team, and organizational levels."

(http://www.sebokwiki.org/) Systems Engineering

Scope of SEBoK Parts and related knowledge



“Systems engineering is an interdisciplinary approach and means to enable the full life cycle of successful product, service and enterprise systems. It including problem discovery and formulation, solution definition and realization, and operational use, sustainment, and disposal. It can be applied to single problem situations or to the management of multiple interventions in commercial or public enterprises.”

http://sebokwiki.org/wiki/Product_System_(glossary)

Product System (glossary)

Guide to the Systems Engineering Body of Knowledge (SEBoK) > enterprise system > enterprise > product system

(1) *A system context for which the system-of-interest (SoI) is a product.* (Created for SEBoK)

(2) *A system considered from the point of view of a physical "system end product" (ANSI/EIA 2003) made of system elements that may include hardware, software, infrastructure and support services. The people and organizational aspects of the "whole system" of which the "product system" forms a part have to be considered in the design, but are provided by another organization.* (ISO/IEC/IEEE 15288 2008)

Sources

- (1) This definition was developed for the SEBoK.
- (2) This definition was extended from: ISO/IEC. 2008. Systems and Software Engineering -- System Life Cycle Processes. Geneva, Switzerland: International Organisation for Standardisation / International Electrotechnical Commissions. ISO/IEC/IEEE 15288:2008.

Discussion

- Definition (1) is has been created for the SEBoK to distinguish from a product from a product system.
- Definition (2) is a systems engineering definition, which is consistent with the concept of a SoI focused on the product system to be delivered to an acquirer who will use it to help deliver user outcomes.
- See the Types of Systems discussion.

"Systems engineering is an interdisciplinary approach and means to enable the full life cycle of successful product, service and enterprise systems. It including problem discovery and formulation, solution definition and realization, and operational use, sustainment, and disposal. It can be applied to single problem situations or to the management of multiple interventions in commercial or public enterprises."

http://sebokwiki.org/wiki/Service_System_(glossary)

Service System (glossary)

Guide to the Systems Engineering Body of Knowledge (SEBoK) > enterprise system > enterprise > product system > service system

> *A dynamic configuration of resources (people, technology, organizations and shared information) that creates and delivers value between the provider and the customer through services.* (IfM and IBM 2008)

Discussion

This dynamic configuration is illustrated in the system coupling diagram (Lawson 2010) where a situation (need for a service) is met by (interacts with) a respondent system (service system) based upon the use of system assets. A service system can also be thought of as a collection of entities that performs the operations, administration, management and provisioning (OAM&P) of resources that together provide the opportunity to co-create value by both the service provider and the service consumer.

The Cambridge white paper defines a service system in this manner (IfM and IBM 2008):

> *Service systems are dynamic configurations of resources (people, technology, organisations and shared information) that can create and deliver service while balancing risk-taking and value co-creation. The dynamics are in part due to the ongoing adjustments and negotiations that occur in all systems involving people. People are the ultimate arbiters of value and risk in service systems (in part because people are legal entities with rights and responsibilities). Service systems are complex adaptive systems.*

Works Cited

- IfM and IBM. 2008. "Succeeding through service innovation: A service perspective for education, research, business and government." University of Cambridge Institute for Manufacturing. Cambridge, UK. cited by Spohrer, J. and P. Maglio. 2010. "Service Science: Toward a Smarter Planet." In *Introduction to Service Engineering*. Ed. G Salvendy and W Karwowski. 3-30. Hoboken, NJ, USA: John Wiley & Sons, Inc.
- Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications, Kings College.

"Systems engineering is an interdisciplinary approach and means to enable the full life cycle of successful product, service and enterprise systems. It including problem discovery and formulation, solution definition and realization, and operational use, sustainment, and disposal. It can be applied to single problem situations or to the management of multiple interventions in commercial or public enterprises."

http://sebokwiki.org/wiki/Enterprise_System_(glossary)

Enterprise System (glossary)

Guide to the Systems Engineering Body of Knowledge (SEBoK) > enterprise system

A system context for which the system-of-interest is an enterprise. (Created for SEBoK)

Source

This definition was developed for the SEBoK.

Discussion

This definition was developed to help distinguish between an enterprise and an enterprise system in the SEBoK. >>>>>>

"Systems engineering is an interdisciplinary approach and means to enable the full life cycle of successful product, service and enterprise systems. It including problem discovery and formulation, solution definition and realization, and operational use, sustainment, and disposal. It can be applied to single problem situations or to the management of multiple interventions in commercial or public enterprises."

http://sebokwiki.org/wiki/Enterprise_(glossary)

Enterprise (glossary)

Guide to the Systems Engineering Body of Knowledge (SEBoK) > enterprise

(1) *one or more organizations sharing a definite mission, goals, and objectives to offer an output such as a product or service.* (ISO 15704 2000)

(2) *An organization (or cross organizational entity) supporting a defined business scope and mission that includes interdependent resources (people, organizations and technologies) that must coordinate their functions and share information in support of a common mission (or set of related missions).* (CIO Council 1999)

(3) *the term enterprise can be defined in one of two ways. The first is when the entity being considered is tightly bounded and directed by a single executive function. The second is when organizational boundaries are less well defined and where there may be multiple owners in terms of direction of the resources being employed. The common factor is that both entities exist to achieve specified outcomes.* (MOD 2004)

(4) *A complex, (adaptive) socio-technical system that comprises interdependent resources of people, processes, information, and technology that must interact with each other and their environment in support of a common mission.* (Giachetti 2010)

Source

(1) ISO 15704. 2000. *Industrial Automation Systems -- Requirements for Enterprise-Reference Architectures and Methodologies*. Geneva, Switzerland: International Organization for Standardization (ISO), ISO 15704:2000.
(2) CIO Council. 1999. *Federal Enterprise Architecture Framework (FEAF)*. Washington, DC, USA: Chief Information Officer (CIO) Council.
(3) MOD. 2004. *Ministry of Defence Architecture Framework (MODAF)*, version 2. London, UK: U.K. Ministry of Defence.
(4) Giachetti, R. E. 2010. *"Design of Enterprise Systems: Theory, Architecture, and Methods."* Boca Raton, FL, USA: CRC Press, Taylor and Francis Group.

Discussion

Definition (1) above is the SEBoK preferred definition. The focus of this definition is on purpose rather than structure, and the enterprise may be either enduring or transient, and either formally or informally constituted and governed. The simplest case of enterprise could be an individual and the largest a national or international undertaking.

Definition (2) is compatible with definition (1) and provides additional insight relevant to the practice of "enterprise architecting", which is closely related to Enterprise Systems Engineering.

Definition (3) highlights the fact that the word "enterprise" is sometimes used synonymously with "business". However in the SEBoK the term business is used to refer to a form of organization, while an enterprise often involves collaboration across organizational boundaries. Furthermore, we also use prodcut and team to refer to two other forms of an organization. An enterprise *could* be at the business level, but some enterprises are at the project or team level. Therefore, as noted above, it is better to focus on the purpose rather than structure of an "enterprise" to facilitate the more proper use and better understanding of that term.

Definition (4) highlights the socio-technical nature of the enterprise and its ability to be adaptable to changing circumstances.

"Systems engineering is an interdisciplinary approach and means to enable the full life cycle of successful product, service and enterprise systems. It including problem discovery and formulation, solution definition and realization, and operational use, sustainment, and disposal. It can be applied to single problem situations or to the management of multiple interventions in commercial or public enterprises."

http://sebokwiki.org/wiki/Enterprise_(glossary)

Enterprise (glossary)

Guide to the Systems Engineering Body of Knowledge (SEBoK) > enterprise

(1) *one or more organizations sharing a definite mission, goals, and objectives to offer an output such as a product or service.* (ISO 15704 2000)

(2) *An organization (or cross organizational entity) supporting a defined business scope and mission that includes interdependent resources (people, organizations and technologies) that must coordinate their functions and share information in support of a common mission (or set of related missions).* (CIO Council 1999)

(3) *the term enterprise can be defined in one of two ways. The first is when the entity being considered is tightly bounded and directed by a single executive function. The second is when organizational boundaries are less well defined and where there may be multiple owners in terms of direction of the resources being employed. The common factor is that both entities exist to achieve specified outcomes.* (MOD 2004)

(4) ***A complex, (adaptive) socio-technical system that comprises interdependent resources of people, processes, information, and technology that must interact with each other and their environment in support of a common mission.*** (Giachetti 2010)

Source

(1) ISO 15704. 2000. *Industrial Automation Systems -- Requirements for Enterprise-Reference Architectures and Methodologies*. Geneva, Switzerland: International Organization for Standardization (ISO), ISO 15704:2000.

(2) CIO Council. 1999. *Federal Enterprise Architecture Framework (FEAF)*. Washington, DC, USA: Chief Information Officer (CIO) Council.

(3) MOD. 2004. *Ministry of Defence Architecture Framework (MODAF)*, version 2. London, UK: U.K. Ministry of Defence.

(4) Giachetti, R. E. 2010. *"Design of Enterprise Systems: Theory, Architecture, and Methods."* Boca Raton, FL, USA: CRC Press, Taylor and Francis Group.

Discussion

Definition (1) above is the SEBoK preferred definition. The focus of this definition is on purpose rather than structure, and the enterprise may be either enduring or transient, and either formally or informally constituted and governed. The simplest case of enterprise could be an individual and the largest a national or international undertaking.

Definition (2) is compatible with definition (1) and provides additional insight relevant to the practice of "enterprise architecting", which is closely related to Enterprise Systems Engineering.

Definition (3) highlights the fact that the word "enterprise" is sometimes used synonymously with "business". However in the SEBoK the term business is used to refer to a form of organization, while an enterprise often involves collaboration across organizational boundaries. Furthermore, we also use prodcut and team to refer to two other forms of an organization. An enterprise *could* be at the business level, but some enterprises are at the project or team level. Therefore, as noted above, it is better to focus on the purpose rather than structure of an "enterprise" to facilitate the more proper use and better understanding of that term.

Definition (4) highlights the socio-technical nature of the enterprise and its ability to be adaptable to changing circumstances.

"(2) *An organization (or cross organizational entity) supporting a defined business scope and mission that includes interdependent resources (people, organizations and technologies) that must coordinate their functions and share information in support of a common mission (or set of related missions).* (CIO Council 1999) …

(4) ***A complex, (adaptive) socio-technical system that comprises interdependent resources of people, processes, information, and technology that must interact with each other and their environment in support of a common mission.*** (Giachetti 2010). "

Definition (2) is compatible with definition (1) and provides additional insight relevant to the practice of "enterprise architecting", which is closely related to Enterprise Systems Engineering…

Definition (4) highlights the socio-technical nature of the enterprise and its ability to be adaptable to changing circumstances"

http://sebokwiki.org/wiki/Enterprise_Architecture_(glossary)

Enterprise Architecture (glossary)

Guide to the Systems Engineering Body of Knowledge (SEBoK) > enterprise > Enterprise Systems Engineering > enterprise > enterprise architecture

(1) *A rigorous description of the structure of an enterprise, its decomposition into subsystems, the relationships between the subsystems, the relationships with the external environment, the terminology to use, and the guiding principles for the design and evolution of an enterprise.* (Giachetti 2009)

(2) *A strategic information asset base, which defines the business, the information necessary to operate the business, the technologies necessary to support the business operations, and the transitional processes necessary for implementing new technologies in response to the changing business needs. It is a representation or blueprint.* (CIO Council 1999)

(3) *The formal description of the structure and function of the components of an enterprise, their interrelationships, and the principles and guidelines governing their design and evolution over time.* (MOD 2004)

(4) *A discipline for proactively and holistically leading enterprise responses to disruptive forces by identifying and analyzing the execution of change toward desired business vision and outcomes. Enterprise architecture delivers value by presenting business and IT leaders with signature-ready recommendations for adjusting policies and projects to achieve target business outcomes that capitalize on relevant business disruptions. It is used to steer decision making toward the evolution of the future state architecture.* (Gartner 2013)

Source

(1) Giachetti, R.E. 2009. *Design of Enterprise Systems: Theory, Architectures, and Methods*. Boca Raton, FL, USA: CRC Press.
(2) CIO Council. 1999. *Federal Enterprise Architecture Framework (FEAF)*. Washington, DC, USA: Chief Information Officer (CIO) Council.
(3) MOD. 2004. *Ministry of Defence Architecture Framework (MODAF)*, version 2. London, UK: U.K. Ministry of Defence.
(4) Gartner IT Glossary. S.V. "enterprise architecture." Accessed 11 March 2013, available at: http://www.gartner.com/it-glossary/enterprise-architecture-ea/.

Discussion

Components of the enterprise can be any element that is a part of the composition of the enterprise and can include people, processes and physical structures, as well as engineering and information systems. (MOD 2004) http://sebokwiki.org/wiki/Enterprise_Systems_Engineering

Enterprise Systems Engineering

Enterprise systems engineering (ESE) is the application of systems engineering principles, concepts, and methods to the planning, design, improvement, and operation of an enterprise.

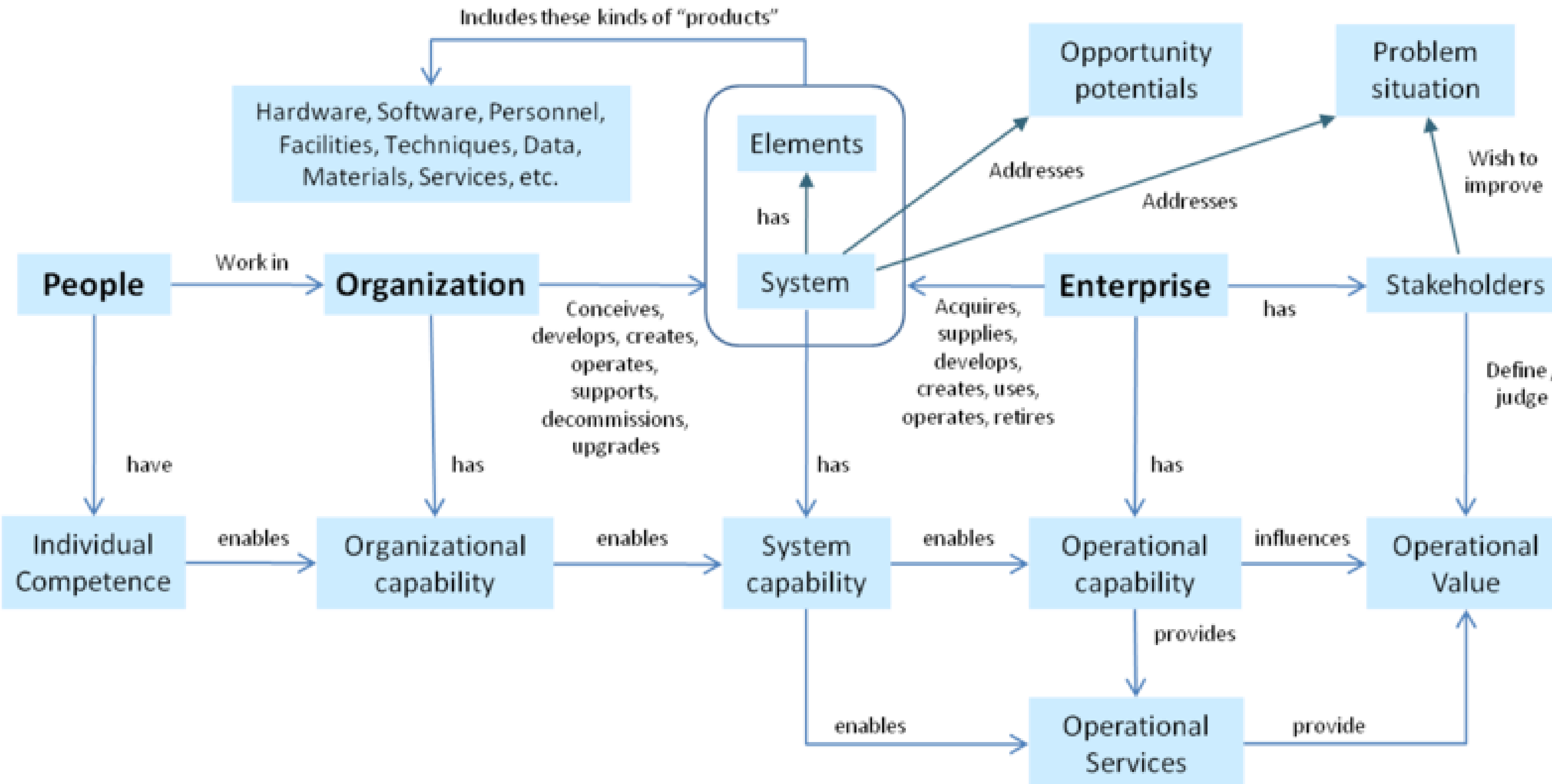
Contents

Topics

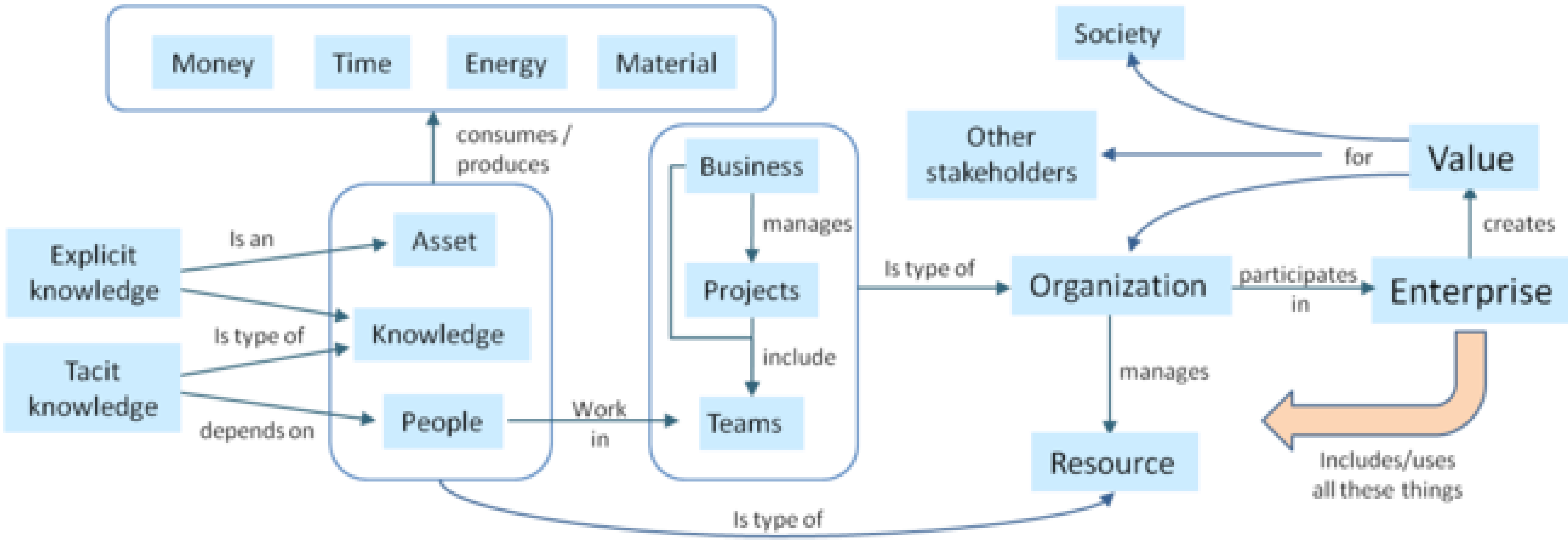
Each part of the SEBoK is divided into knowledge areas (KAs), which are groupings of information with a related theme. The KAs in turn are divided into topics. This KA contains the following topics:

- Enterprise Systems Engineering Background
- The Enterprise as a System
- Related Business Activities
- Enterprise Systems Engineering Key Concepts
- Enterprise Systems Engineering Process Activities
- Enterprise Capability Management

http://sebokwiki.org/wiki/Enterprise_Systems_Engineering_Background



http://sebokwiki.org/wiki/The_Enterprise_as_a_System

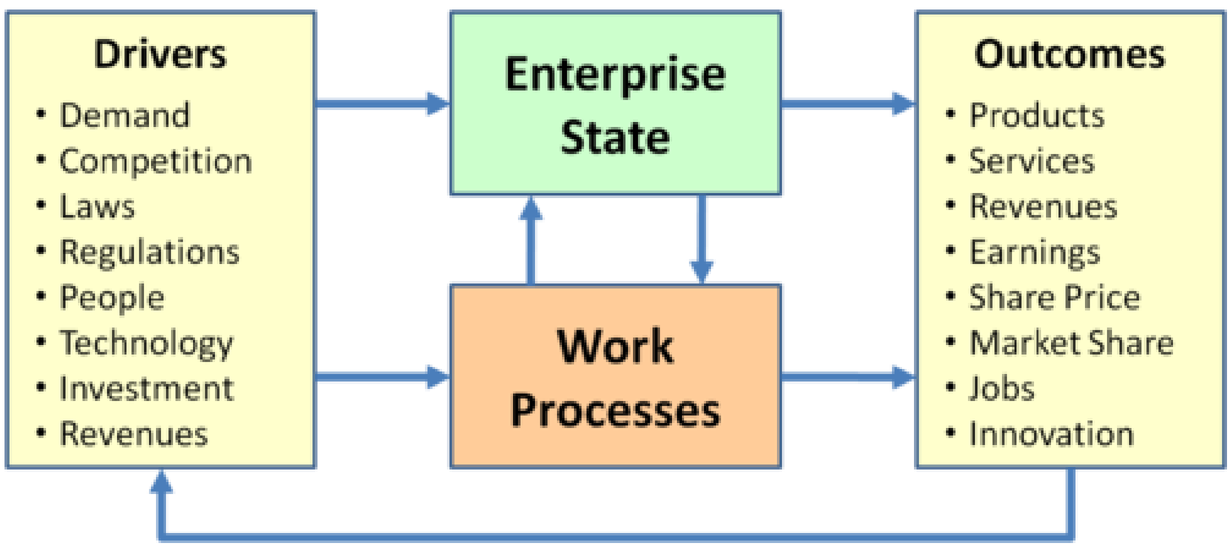


Notes:

1. All entities shown are decomposable, except people. For example, a business can have sub-businesses, a project can have subprojects, a resource can have sub-resources, an enterprise can have sub-enterprises.
2. All entities have other names. For example, a program can be a project comprising several subprojects (often called merely projects). Business can be an agency, team can be group, value can be utility, etc.
3. There is no attempt to be prescriptive in the names chosen for this diagram. The main goal of this is to show how this chapter uses these terms and how they are related to each other in a conceptual manner.

http://sebokwiki.org/wiki/Enterprise_Systems_Engineering_Background

Models of the enterprise can serve as the basis for understanding the enterprise in its context of markets and economies. The figure below shows the various drivers (or inputs) of an enterprise and its potential outcomes (or outputs) (Rouse 2009). Enterprise architecture can be a key enabler for modeling and can serve as a basis for transformation (Vernadat 1996; Bernus, Laszlo, and Schmidt 2003; Nightingale and Rhodes 2004). Enterprise architecture can be used to provide a model to understand how the parts of the enterprise fit together (or do not) (Giachetti 2010) (See also Representing Systems with Models). For a good review of the subject see Lillehagen and Krogstie (2008).



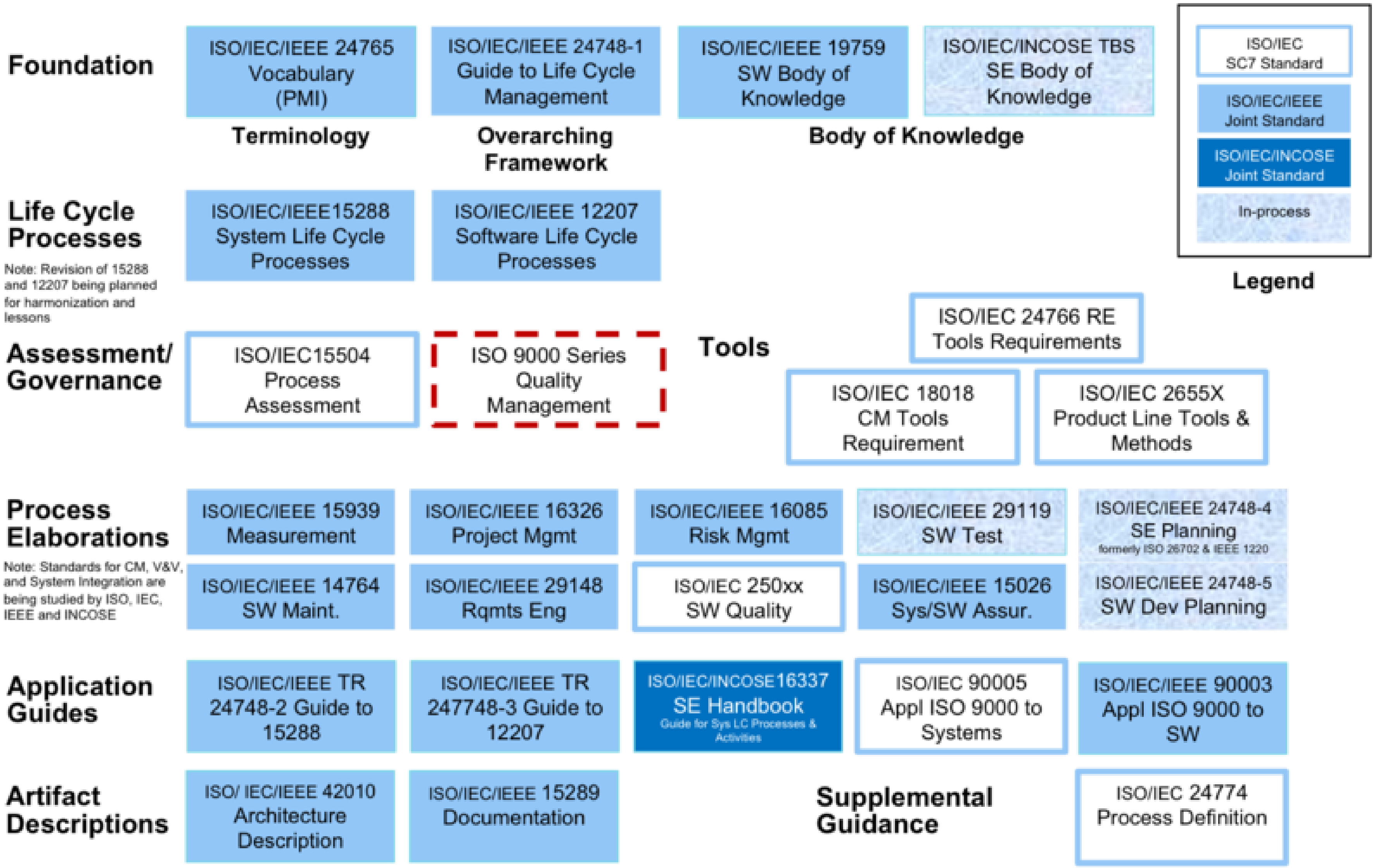
http://sebokwiki.org/wiki/Relevant_Standards

Standard Type	Description of Type
Concepts and Terminology	• Defines the terminology and describes the concepts of a specific domain.
Process	• Elaborates a specific process, giving normative requirements for the essential elements of the process. It may give guidance to the requirements.
Requirements	• Describes the requirements for something. • Most often used for actions, activities, or practices and not objects (see specifications).
Procedure (Practice, Activity)	• A specific procedure. Instructions or requirements on how to do something. • Could be a description of best practices. • Sometimes guidance and sometimes normative.
Guidance	• Usually an interpretation and guidance of a published standard.
Management System	• Requirements for management.
Specification	• Specifies the form, attributes, or properties of a subject artifact. • Usually an object and usually normative.
Reference Model	• A reference model or collection of specifications of which a reference model is composed.
Process Reference Model (PRM)	• A collection of processes necessary and sufficient to achieve a nominated business outcome.
Process Assessment Model (PAM)	• Requirements and guidance for assessing attributes of nominated processes or attributes of a nominated collection of processes.
Guide to Body of Knowledge (BOK)	• Collects and describes the current body of knowledge in a domain, or guidance to the body of knowledge.

http://sebokwiki.org/wiki/Relevant_Standards (just for curiosity...)

Document ID	Document Title	Organization
ISO/IEC/IEEE 15288	Systems and Software Engineering - System Life Cycle Processes	ISO/IEC/IEEE
ISO/IEC/IEEE 24765	Systems and Software Engineering - Systems and Software Engineering Vocabulary	ISO/IEC/IEEE
ISO/IEC/IEEE 42010	Systems and Software Engineering - Architecture Description	ISO/IEC/IEEE
ISO/IEC 26702 / IEEE 1220	Management of the Systems Engineering Process	ISO/IEC/IEEE
ISO/IEC/IEEE 29148	Systems and Software Engineering - Requirements Engineering	ISO/IEC/IEEE
ISO/IEC/IEEE 16085	Systems and Software Engineering - Risk Management	ISO/IEC/IEEE
ISO/IEC/IEEE 15939	Systems and Software Engineering - Measurement Process	ISO/IEC/IEEE
ISO/IEC/IEEE 16326	Systems and Software Engineering - Project Management	ISO/IEC/IEEE
prEN9277	Programme management - Guide for the management of Systems Engineering	CEN
EIA 632	Engineering of a System	TechAmerica
ISO 9001:2008	Quality Management Systems - Requirements	ISO TC 176
EIA-649-B	National Consensus Standard for Configuration Management	TechAmerica
ISO/IEC/IEEE TR 24748-1	Systems and Software Engineering - Guide to Life Cycle Management	ISO/IEC/IEEE
ISO/IEC/IEEE TR 24748-2	Systems and Software Engineering - Guide To The Application of ISO/IEC 15288:2008	ISO/IEC/IEEE
ISO/IEC/IEEE CD 24748-4	Systems and Software Engineering - Application and management of the systems engineering process	ISO/IEC/IEEE
ISO/IEC DTR 16337	Systems Engineering Handbook (INCOSE)	ISO/IEC/INCOSE
ISO/IEC/IEEE 15289:2011	Systems and Software Engineering - Content of Life-Cycle Information Products (Documentation)	ISO/IEC/IEEE
ISO/IEC/IEEE 15026-1:2010	Systems and Software Engineering - System and Software Assurance – Part 1: Concepts And Vocabulary	ISO/IEC/IEEE
ISO/IEC/IEEE 15026-2:2011	Systems and Software Engineering - System and Software Assurance – Part 2: Assurance Case	ISO/IEC/IEEE
ISO/IEC/IEEE 15026-3:2011	Systems and Software Engineering - System and Software Assurance – Part 3: Integrity Levels	ISO/IEC/IEEE
ISO/IEC/IEEE 15026-4:2012	Systems and Software Engineering - System And Software Assurance – Part 4: Assurance in the Life Cycle	ISO/IEC/IEEE JTC 1
ISO/IEC TR 90005:2008	Guidelines for the Application of ISO 9001 to Systems Life Cycle Processes	ISO/IEC JTC 1
ISO 10303-233:2012	Systems Engineering Data Interchange Standard	ISO TC 184
ECSS-E-ST-10C	Systems Engineering General Requirements	ECSS
ECSS-E-ST-10-02	Space Engineering - Verification {Note - standard is canceled}	ECSS
ECSS-E-ST-10-06	Space Engineering - Technical Requirements Specification	ECSS
ECSS-E-ST-10-24	Space Engineering - Interface Control	ECSS
ECSS-M-ST-10	Space Project Management - Project Planning and Implementation	ECSS
ECSS-M-ST-40	Space Project Management - Configuration and Information Management	ECSS
ECSS-M-00-03	Space Project Management - Risk Management	ECSS
ISO 31000:2009	Risk Management - Principles and Guidelines	ISO
ISO 31010:2009	Risk Management - Risk Assessment Techniques	ISO
ISO 19439:2006	Enterprise Integration - Framework for Enterprise Modeling	ISO
ISO 15704:2000	Requirements for Enterprise - Reference Architectures and Methodologies	ISO
EIA 748	Earned Value Management System	TechAmerica

http://sebokwiki.org/wiki/Alignment_and_Comparison_of_the_Standards



http://sebokwiki.org/wiki/Systems_Engineering_and_Management

The commonly recognized definition of systems engineering (SE) used across the SEBoK (INCOSE 2015) defines SE as an interdisciplinary approach which applies across the complete life cycle of an identified System-of-Interest. The definition states that systems engineering "**integrates all the disciplines and speciality groups into a team effort forming a structured development process that proceeds from concept to production to operation**". Thus, SE is an engineering discipline concerned with all aspects of an engineered systems life, including how we organize to do the engineering, what is produced by that engineering and how the resulting systems are used and sustained to meet stakeholder needs.

Systems engineering is transitioning to a model-based approach, model-based systems engineering (MBSE), like many other engineering disciplines. The aim is to enhance productivity and quality, and to cope with the design of increasingly complex systems. Although, models have always been used by systems engineering to create information about engineered systems, that information has been translated and managed through document based artifacts. In a model-based approach, the information about the system is captured in a shared system model, made up of a set of integrated models appropriate to the life cycle stages. This model is managed and controlled throughout the system life cycle as noted in Part 2 under Representing Systems with Models.

http://sebokwiki.org/wiki/What_is_a_Model%3F

There are many definitions of the word *model*. The following definitions refer to a model as a representation of selected aspects of a domain of interest to the modeler:

- a physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process (DoD 1998);
- a representation of one or more concepts that may be realized in the physical world (Friedenthal, Moore, and Steiner 2009);
- a simplified representation of a system at some particular point in time or space intended to promote understanding of the real system (Bellinger 2004);
- an abstraction of a system, aimed at understanding, communicating, explaining, or designing aspects of interest of that system (Dori 2002); and
- a selective representation of some system whose form and content are chosen based on a specific set of concerns; the model is related to the system by an explicit or implicit mapping (Object Management Group 2010).

In the context of systems engineering, a model that represents a system and its environment is of particular importance to the system engineer who must specify, design, analyze, and verify systems, as well as share information with other stakeholders. A variety of system models are used to represent different types of systems for different modeling purposes…

http://sebokwiki.org/wiki/Why_Model%3F

Purpose of a Model

Models are representations that can aid in defining, analyzing, and communicating a set of concepts. System models are specifically developed to support analysis, specification, design, verification, and validation of a system, as well as to communicate certain information. One of the first principles of modeling is to clearly define the purpose of the model. Some of the purposes that models can serve throughout the system life cycle are:

- **Characterizing an existing system:** Many existing systems are poorly documented, and modeling the system can provide a concise way to capture the existing system design. ...
- **Mission and system concept formulation and evaluation:** Models can be applied early in the system life cycle to synthesize and evaluate alternative mission and system concepts. This includes clearly and unambiguously defining the system's mission and the value it is expected to deliver to its beneficiaries. Models can be used to explore a trade-space by modeling alternative system designs and assessing the impact of critical system parameters such as weight, speed, accuracy, reliability, and cost on the overall measures of merit. In addition to bounding the system design parameters, models can also be used to validate that the system requirements meet stakeholder needs before proceeding with later life cycle activities such as synthesizing the detailed system design.
- **System design synthesis and requirements flowdown:** Models can be used to support architecting system solutions, as well as flow mission and system requirements down to system components. Different models may be required to address different aspects of the system design and respond to the broad range of system requirements. This may include models that specify functional, interface, performance, and physical requirements, as well as other non-functional requirements such as reliability, maintainability, safety, and security.
- **Support for system integration and verification:** Models can be used to support integration of the hardware and software components into a system, as well as to support verification that the system satisfies its requirements. This often involves integrating lower level hardware and software design models with system-level design models which verify that system requirements are satisfied. System integration and verification may also include replacing selected hardware and design models with actual hardware and software products in order to incrementally verify that system requirements are satisfied. This is referred to as hardware-in-the-loop testing and software-in-the-loop testing. Models can also be used to define the test cases and other aspects of the test program to assist in test planning and execution.
- **Support for training:** Models can be used to simulate various aspects of the system to help train users to interact with the system. Users may be operators, maintainers, or other stakeholders. …
- **Knowledge capture and system design evolution:** Models can provide an effective means for capturing knowledge about the system and retaining it as part of organizational knowledge. This knowledge, which can be reused and evolved, provides a basis for supporting the evolution of the system, such as changing system requirements in the face of emerging, relevant technologies, new applications, and new customers. …

http://sebokwiki.org/wiki/Why_Model%3F

Indicators of an Effective Model

When modeling is done well, a model's purposes are clear and well-defined. The value of a model can be assessed in terms of how effectively it supports those purposes. …

Model Scope

The model must be scoped to address its intended purpose. In particular, the types of models and associated modeling languages selected must support the specific needs to be met. For example, suppose models are constructed to support an aircraft's development. …

For each type of model, the appropriate breadth, depth, and fidelity should be determined to address the model's intended purpose. The model breadth reflects the system requirements coverage in terms of the degree to which the model must address the functional, interface, performance, and physical requirements, as well as other non-functional requirements, such as reliability, maintainability, and safety. ...

The model's depth indicates the coverage of system decomposition from the system context down to the system components.

Indicators of Model Quality

The quality of a model should not be confused with the quality of the design that the model represents. For example, one may have a high-quality, computer-aided design model of a chair that accurately represents the design of the chair, yet the design itself may be flawed such that when one sits in the chair, it falls apart. A high quality model should provide a representation sufficient to assist the design team in assessing the quality of the design and uncovering design issues.

Model quality is often assessed in terms of the adherence of the model to modeling guidelines and the degree to which the model addresses its intended purpose. Typical examples of modeling guidelines include naming conventions, application of appropriate model annotations, proper use of modeling constructs, and applying model reuse considerations. Specific guidelines are different for different types of models. For example, the guidelines for developing a geometric model using a computer-aided design tool may include conventions for defining coordinate systems, dimensioning, and tolerances.

Model-based Metrics

…

http://sebokwiki.org/wiki/System_Modeling_Concepts

Abstraction

Perhaps the most fundamental concept in systems modeling is abstraction, which concerns hiding unimportant details in order to focus on essential characteristics. Systems that are worth modeling have too many details for all of them to reasonably be modeled. Apart from the sheer size and structural complexity that a system may possess, a system may be behaviorally complex as well, with emergent properties, non-deterministic behavior, and other difficult-to-characterize properties. Consequently, models must focus on a few vital characteristics in order to be computationally and intellectually tractable. Modeling techniques address this complexity through various forms of abstraction. For example, a model may assume that structural characteristics of many individual components of a particular type are all the same, ignoring the small order differences between individuals in instances that occur in real life. In that case, those differences are assumed to be unimportant to modeling the structural integrity of those components. Of course, if that assumption is wrong, then the model could lead to false confidence in that structural integrity. There are two key concepts that are applied in regard to modeling different levels of abstraction, which are: view and viewpoint and black-box and white-box modeling, which are described below. Although these two modeling methods are the most widely recognized, different modeling languages and tools employ other techniques as well.

http://sebokwiki.org/wiki/System_Modeling_Concepts

View and Viewpoint

IEEE 1471, a standard for architecture modeling, defines "view" and "viewpoint" as follows:

- view - A representation of a whole system from the perspective of a related set of concerns.
- viewpoint - A specification of the conventions necessary for constructing and using a view; a pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis.

Even though IEEE 1471 is focused on architecture models, the concepts of view and viewpoint are general and could apply to models for other purposes as well (IEEE 2000). The viewpoint addresses the concerns of the stakeholders and provides the necessary conventions for constructing a view to address those concerns; therefore, the view represents aspects of the system that address the concerns of the stakeholder. Models can be created to represent the different views of the system. A systems model should be able to represent multiple views of the system to address a range of stakeholder concerns. Standard views may include requirements, functional, structural, and parametric views, as well as a multitude of discipline-specific views to address system reliability, safety, security, and other quality characteristics.

Black-Box and White-Box Models

A very common abstraction technique is to model the system as a black-box, which only exposes the features of the system that are visible from an external observer and hides the internal details of the design. This includes externally visible behavior and other physical characteristics, such as the system’s mass or weight. A white-box model of a system, on the other hand, shows the internal structure and displays the behavior of the system. Black-box and white-box modeling can be applied to the next level of design decomposition in order to create a black-box and white-box model of each system component.

http://sebokwiki.org/wiki/System_Modeling_Concepts

Conceptual Model

A conceptual model is the set of concepts within a system and the relationships among those concepts (e.g., view and viewpoint). A system conceptual model describes, using one diagram type (such as in Object-Process Methodology (OPM)) or several diagram types (such as in Systems Modeling Language (SysML)) the various aspects of the system. The conceptual model might include its requirements, behavior, structure, and properties. In addition, a system conceptual model is accompanied by a set of definitions for each concept. Sometimes, system concept models are defined using an entity relationship diagram, an object-process diagram (OPD), or a Unified Modeling Language (UML) class diagram.

A preliminary conceptual (or concept) model for systems engineering (Systems Engineering Concept Model) was developed in support of the integration efforts directed toward the development of the Object Management Group (OMG) SysML and the International Organization for Standardization (ISO) AP233 *Data Exchange Standard for Systems Engineering* (ISO 2010). The concept model was originally captured in an informal manner; however, the model and associated concepts were rigorously reviewed by a broad representation of the systems engineering community, including members from the International Council on Systems Engineering (INCOSE), AP233, and SysML development teams.

A fragment from the top level systems engineering concept model is included in Figure 1. This model provides concepts for requirements, behavior, structure and properties of the system, as well as other concepts common to systems engineering and project management, such as stakeholder. The concept model is augmented by a well-defined glossary of terms called the semantic dictionary. …

http://sebokwiki.org/wiki/Modeling_Standards

Descriptive Models - These standards apply to general descriptive modeling of systems:

- Functional Flow Block Diagram (FFBD) (Oliver, Kelliher, and Keegan 1997)
- Integration Definition for Functional Modeling (IDEF0) (NIST 1993)
- Object-Process Methodology (OPM) [[1]] [[2]] (Dori 2002; ISO 19450 PAS - Publicly Available Specification in progress)
- **Systems Modeling Language (SysML)**(OMG 2010a)
- Unified Profile for United States Department of Defense Architecture Framework (DoDAF) and United Kingdom Ministry of Defense Architecture Framework (MODAF) (OMG 2011e)
- Web ontology language (OWL) (W3C 2004b)

Analytical Models and Simulations - These standards apply to analytical models and simulations:

- Distributed Interactive Simulation (DIS) (IEEE 1998)
- High-Level Architecture (HLA) (IEEE 2010)
- Modelica (Modelica Association 2010)
- Semantics of a Foundational Subset for Executable Unified Modeling Language (UML) Models (FUML) (OMG 2011d)

Data Exchange Standards - These standards enable the exchange of information between models:

- Application Protocol for Systems Engineering Data Exchange (ISO 10303-233) (AP-233) (ISO 2005)
- Requirements Interchange Format (ReqIF) (OMG 2011c)
- Extensible Mark-Up Language -(XML) Metadata Interchange (XMI) (OMG 2003a)
- Resource Description Framework (RDF) (W3C 2004a)

Model Transformations - These standards apply to transforming one model to another to support semantic interoperability:

- Query View Transformations (QVT) (OMG 2011b)
- Systems Modeling Language (SysML)-Modelica Transformation (OMG 2010c)
- OPM-to-SysML Transformation (Grobshtein and Dori 2011)

General Modeling Standards - These standards provide general frameworks for modeling:

- Model-driven architecture (MDA®) (OMG 2003b)
- IEEE 1471-2000 - Recommended Practice for Architectural Description of Software-Intensive Systems (ANSI/IEEE 2000) (ISO/IEC 2007)

Other Domain-Specific Modeling Standards

Software Design Models - These standards apply to modeling application software and/or embedded software design:

- Architecture Analysis and Design Language (AADL) (SAE 2009)
- Modeling and Analysis for Real-Time and Embedded Systems (MARTE) (OMG 2009)
- **Unified Modeling Language (UML) (OMG 2010b)**

Hardware Design Models - These standards apply to modeling hardware design:

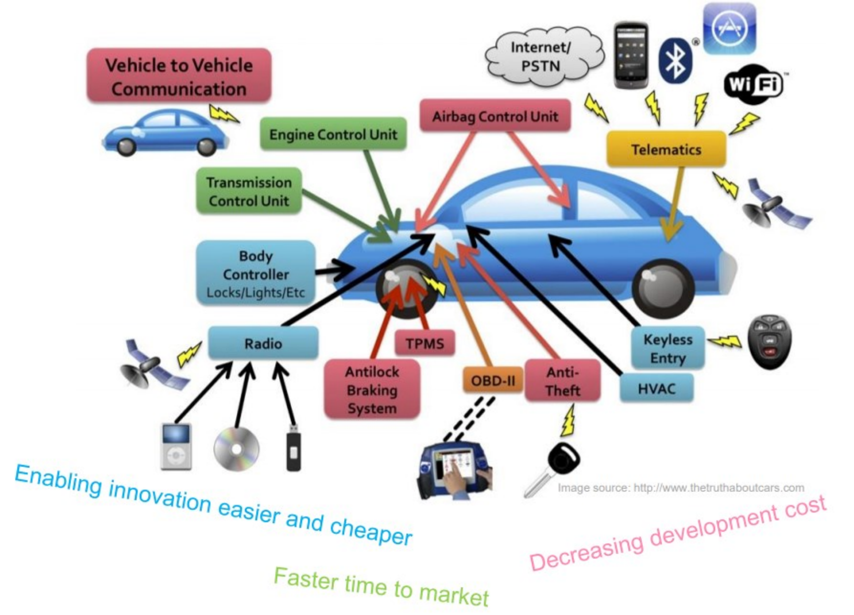
- Very-High-Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) (IEEE 2008)

Business Process Models - These standards apply to modeling business processes:

- **Business Process Modeling Notation (BPMN) (OMG 2011a)**

Enterprise Architecure

- **The ArchiMate Enterprise Architecture Modeling Language**



AMS

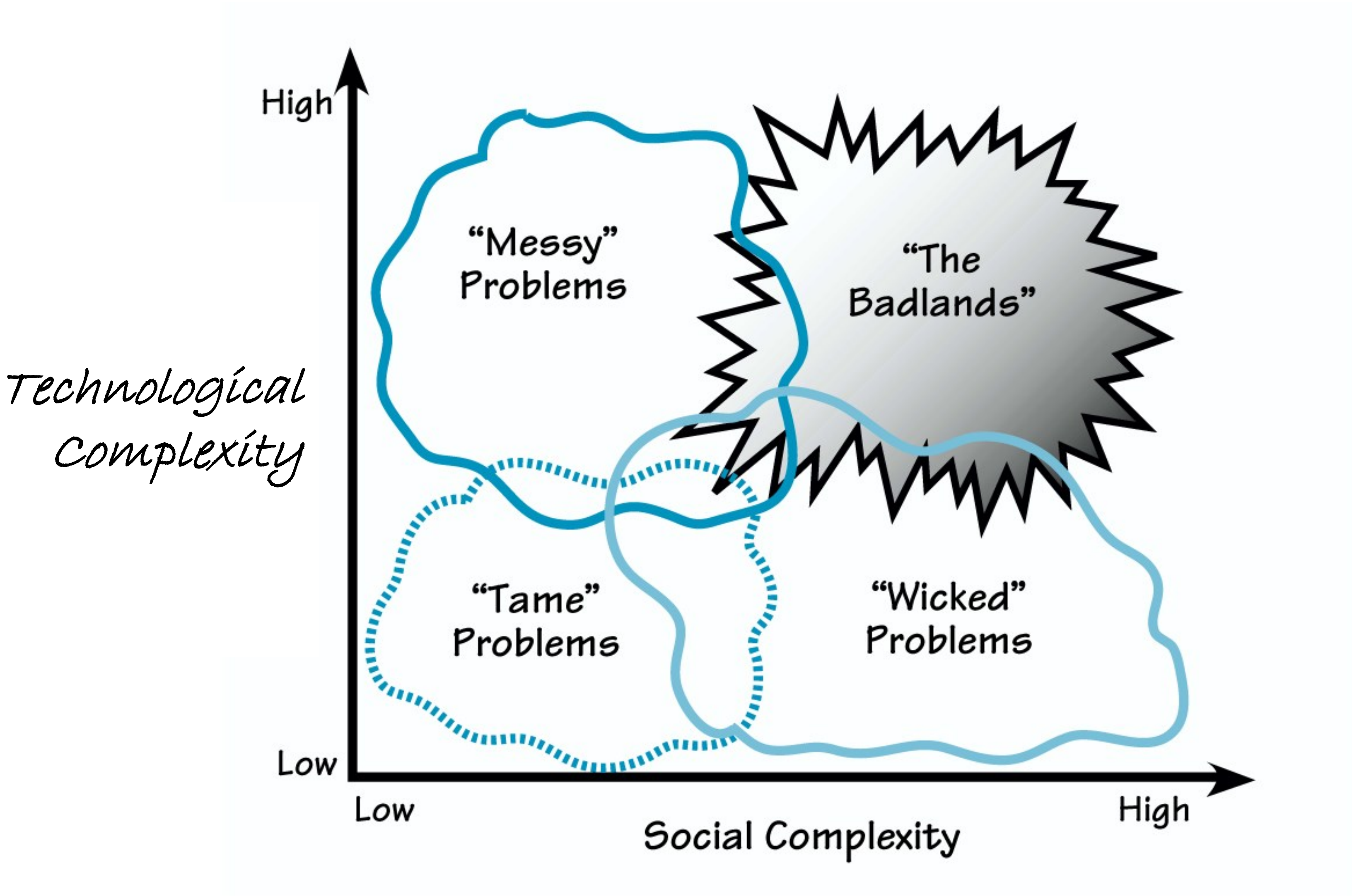
A very short address of

Socio-technical Systems…,

Systems of Systems,…

Information Systems,…

When we also have "people in the loop", we have a "socio-technical" system (and it all can be very hard to address… but we always must try our best ;-)



Adapted from https://thesystemsthinker.com/learning-and-leading-through-the-badlands/

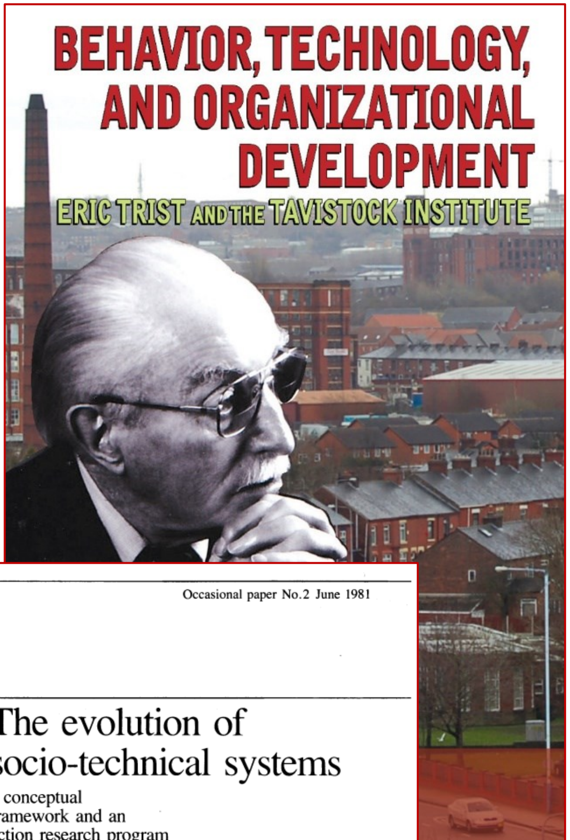
What about Information Systems?

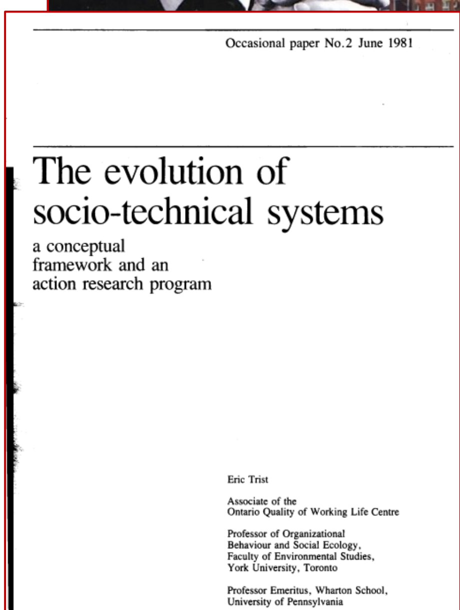
Information system: an information processing system, together with associated organizational resources such as human, technical, and financial resources, that provides and distributes information.
(ISO/IEC 2382-1:1993 Information technology--Vocabulary--Part 1: Fundamental terms, 01.01.22)

Or, let us agree here in our definition:

An information system is a group of socio-technical entities (technological and human entities) interacting in a coordinated way with the purpose of satisfying information needs of an organization.

Socio-Technical Systems?




Occasional paper No.2 June 1981

The evolution of socio-technical systems

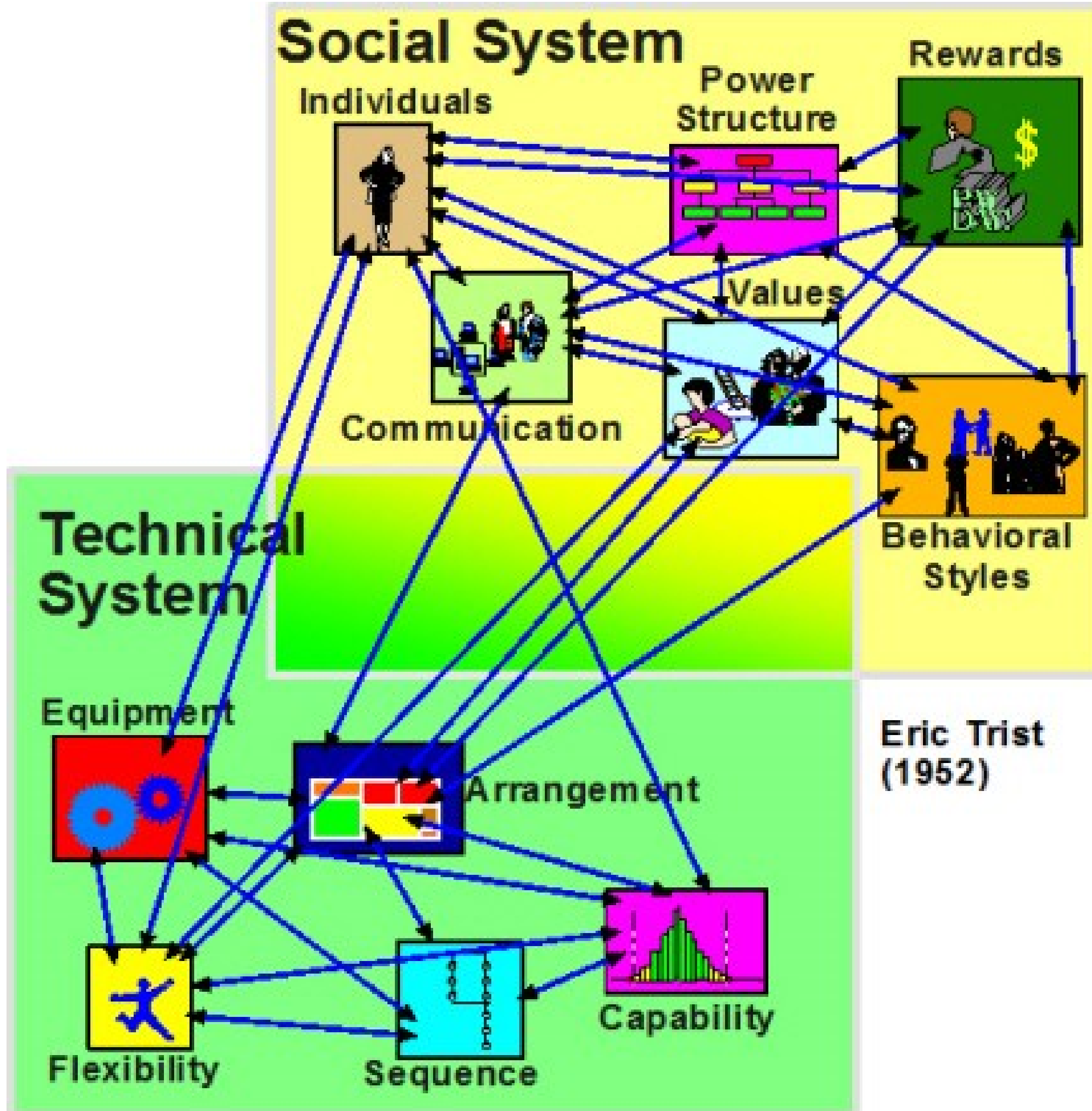
a conceptual framework and an action research program

Eric Trist

Associate of the
Ontario Quality of Working Life Centre

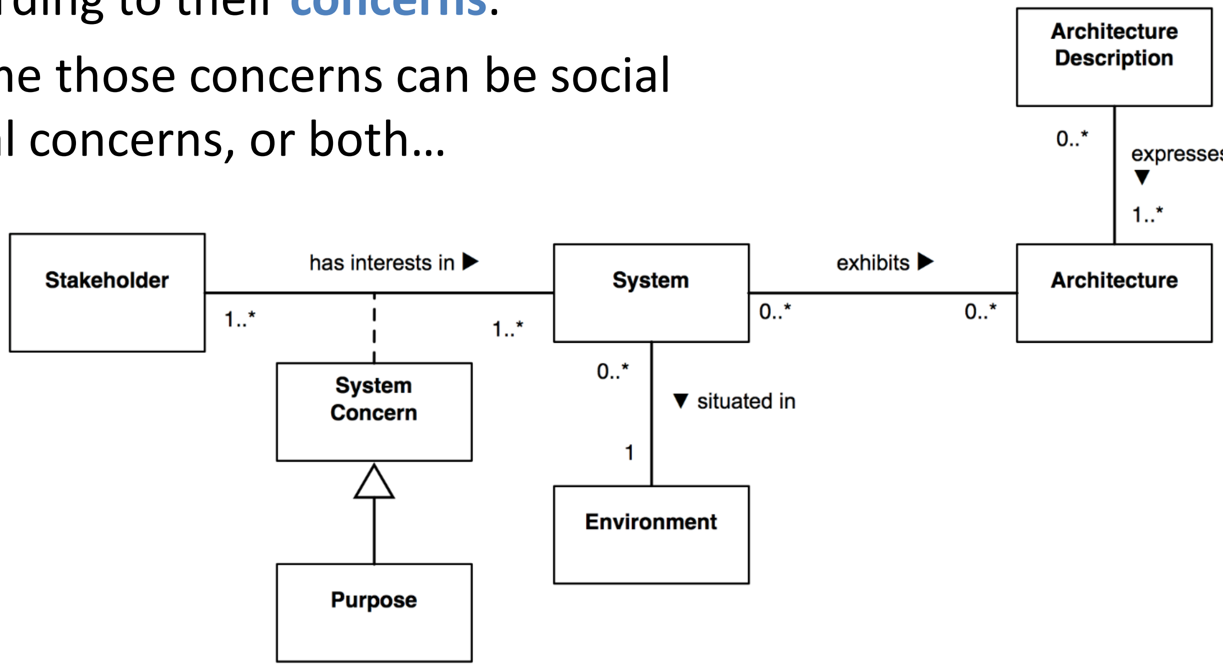
Professor of Organizational
Behaviour and Social Ecology,
Faculty of Environmental Studies,
York University, Toronto

Professor Emeritus, Wharton School,
University of Pennsylvania



Back to our discussion on "system"

- "A **system** is a **thing** that when situated in a specific **environment** **pursue a common goal** for a set of **stakeholders** according to their **concerns**."
- Plus, we will assume those concerns can be social concerns, technical concerns, or both…



Concluding: In this course we'll understand a system as a **thing** made of a **set of entities** that can be software, hardware, people, or any other unit or any combination thereof…

Thus, what is our role in all of this, as engineers, and form the perspective of this course?

An information system is a group of socio-technical entities (technological and human entities) interacting in a coordinated way with the purpose of satisfying information needs of an organization.

An information system is...

a) ...a group of socio-technical entities (technological and human entities)

b) ...interacting in a coordinated way

c) ...with the purpose of satisfying information needs of an organization.

1) Find the information needs...

2) ...Find the related entities...

3) ...Understand how these entities must be “grouped”...

4) ...Understand how that group must be “coordinated”

Thus, what is our role in all of this, as engineers, and form the perspective of this course?

An information system is a group of socio-technical entities (technological and human entities) interacting in a coordinated way with the purpose of satisfying information needs of an organization.

An information system is…

a) …a group of socio-technical entities (technological and human entities)

b) …interacting in a coordinated way

c) …with the purpose of satisfying information needs of an organization.

1) Find the information needs…

2) …Find the related entities…

3) …Understand whose of these entities must be “grouped” and how…

4) …Understand how that group must be “coordinated”

Requirements

Context

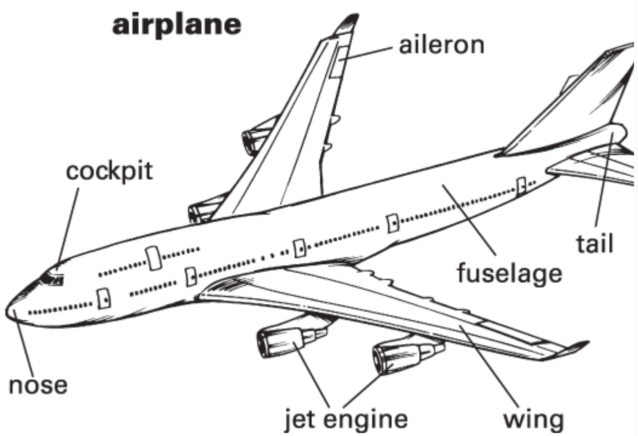
Structure

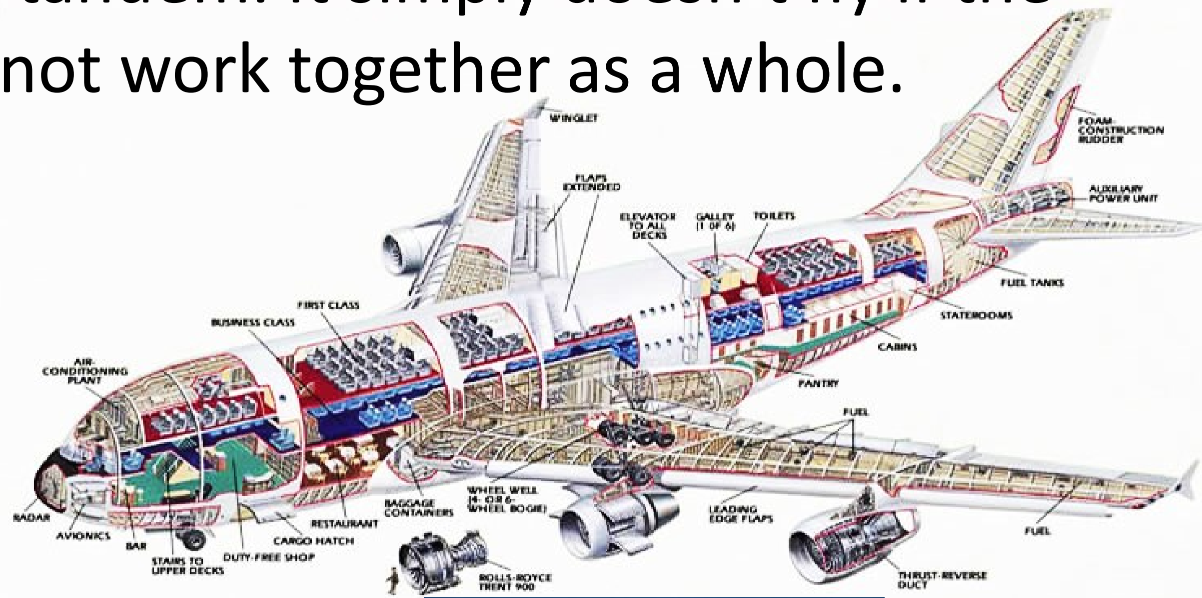
Behaviour

A tale of when a complex socio-technical system went wrong… ☹

System of Systems (SoS)

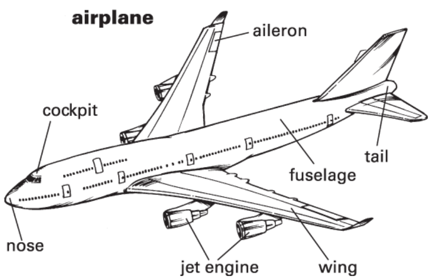
Many **systems** operate various parts of a **plane** (which itself is also a system), but the plane only flies when all its systems work in tandem. It simply doesn't fly if the systems (parts) do not work together as a whole.

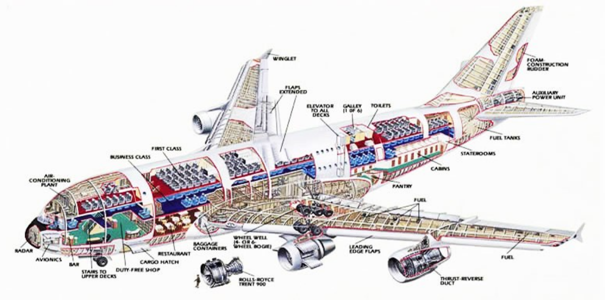


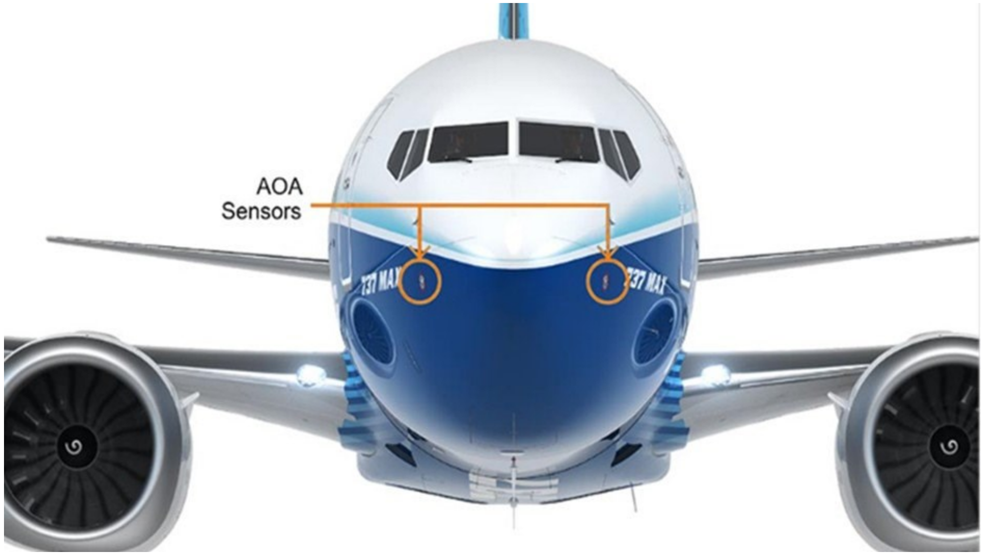


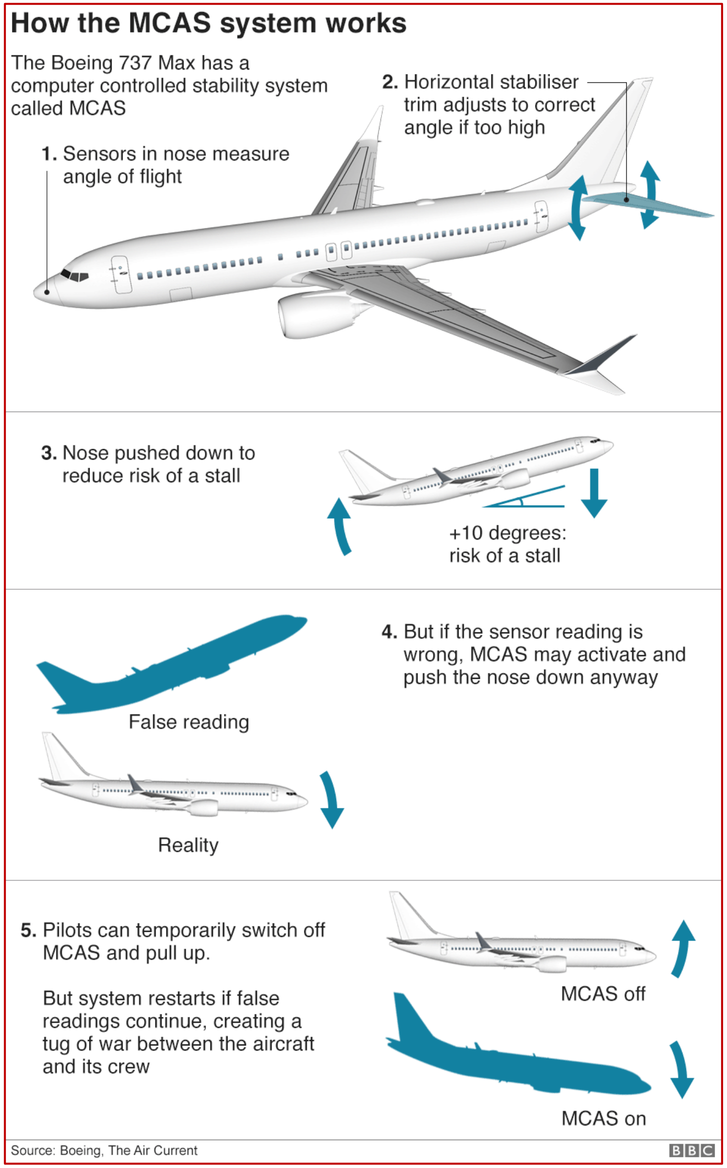
Systems of Systems (SoS) that go wrong ☹

Many **systems** operate various parts of a **plane** (which itself is also a system), but the plane only flies when all its systems work in tandem. It simply doesn't fly if the systems (parts) do not work together as a whole.

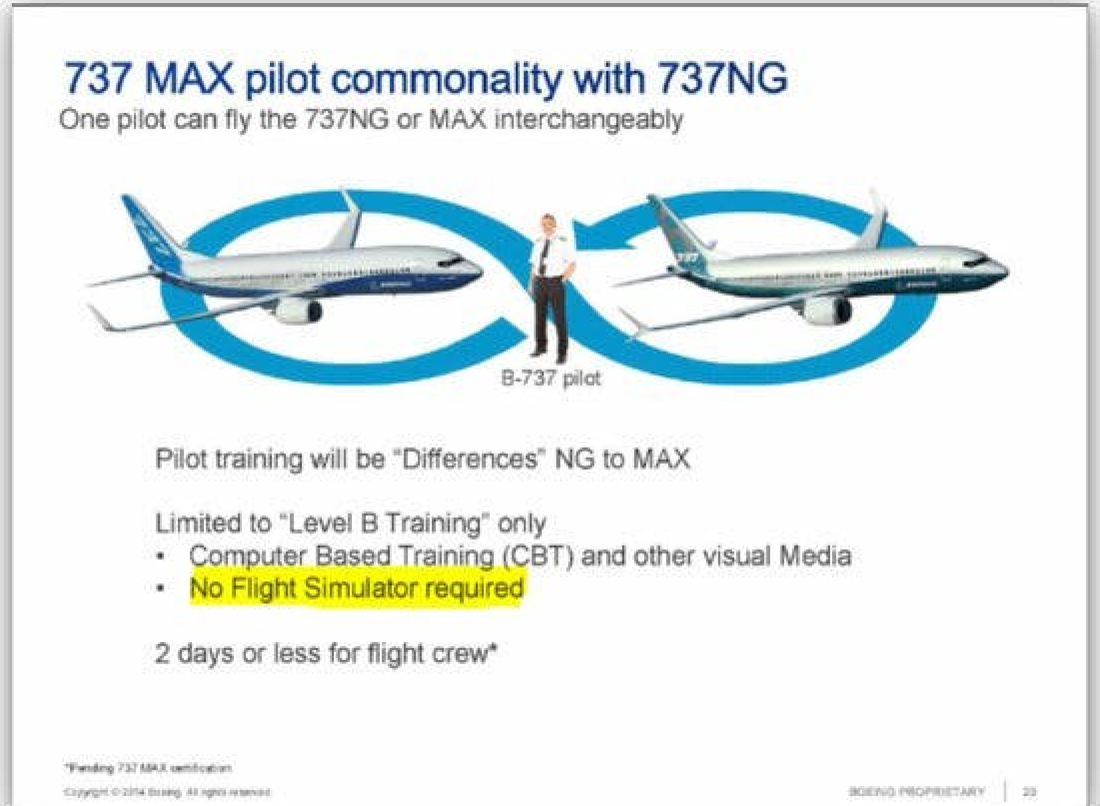






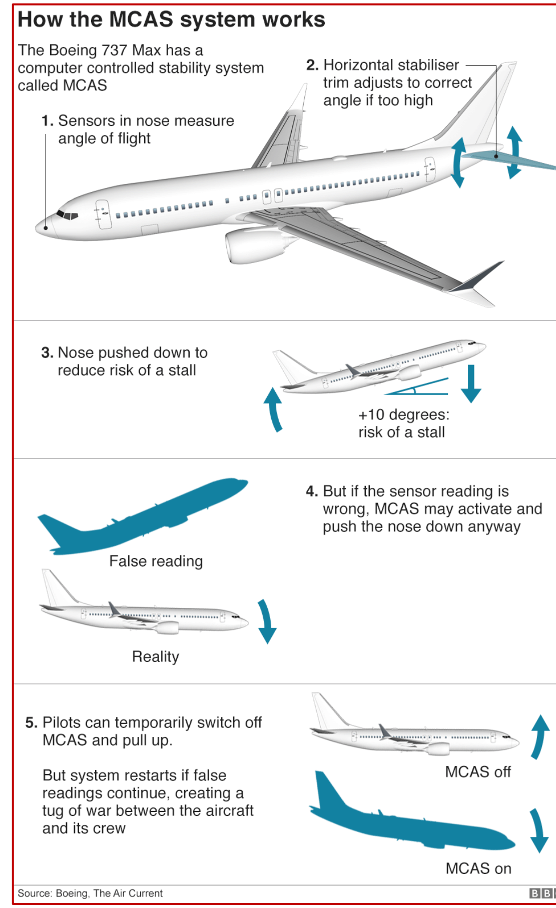


Systems of Systems (SoS) that go wrong ☹



Boeing's marketing material for the 737 Max promoted the fact that simulator training wouldn't be needed.

https://www.nytimes.com/2020/01/07/business/boeing-737-max-simulator-training.html



A flying airplane is indeed a socio-technical system!!!


737 MAX pilot commonality with 737NG
One pilot can fly the 737NG or MAX interchangeably

B-737 pilot

Pilot training will be "Differences" NG to MAX

Limited to "Level B Training" only
- Computer Based Training (CBT) and other visual Media
- No Flight Simulator required

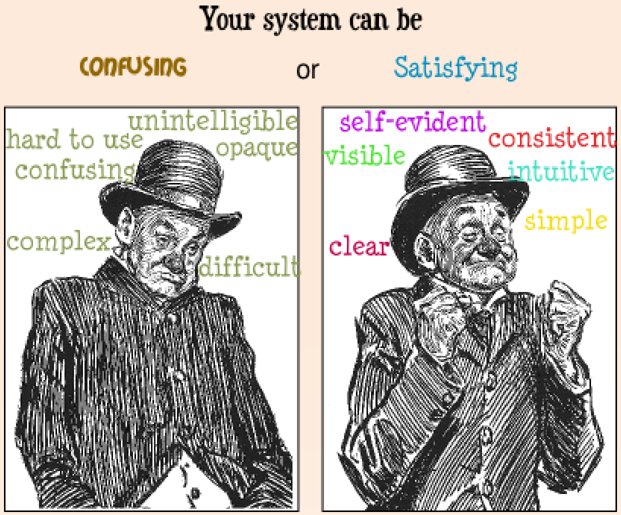
2 days or less for flight crew*

https://www.nytimes.com/2020/01/07/business/boeing-737-max-simulator-training.html
https://www.latimes.com/business/story/2020-01-09/boeing-737-max-simulators

http://www.lispcast.com/wp-content/uploads/2008/01/good-and-bad-conceptual-models.png

AMS

Conceptual Models…

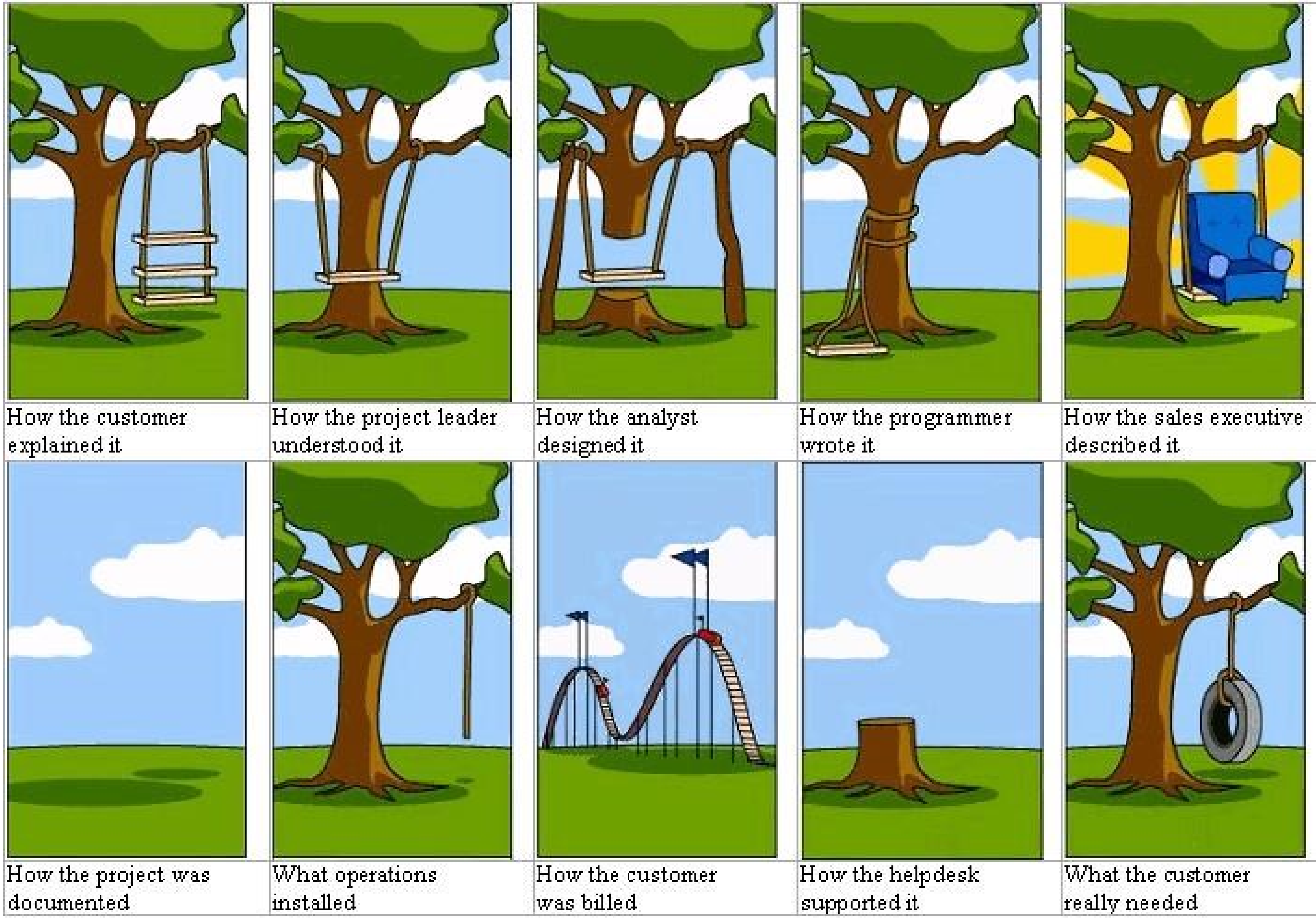


"Tout ce que nous voyons cache quelque chose d'autre."
René Magritte (1898-1967)

"Everything we see, hides something else (…that we are not seeing…)."
René Magritte…

The Treachery of Images, 1929

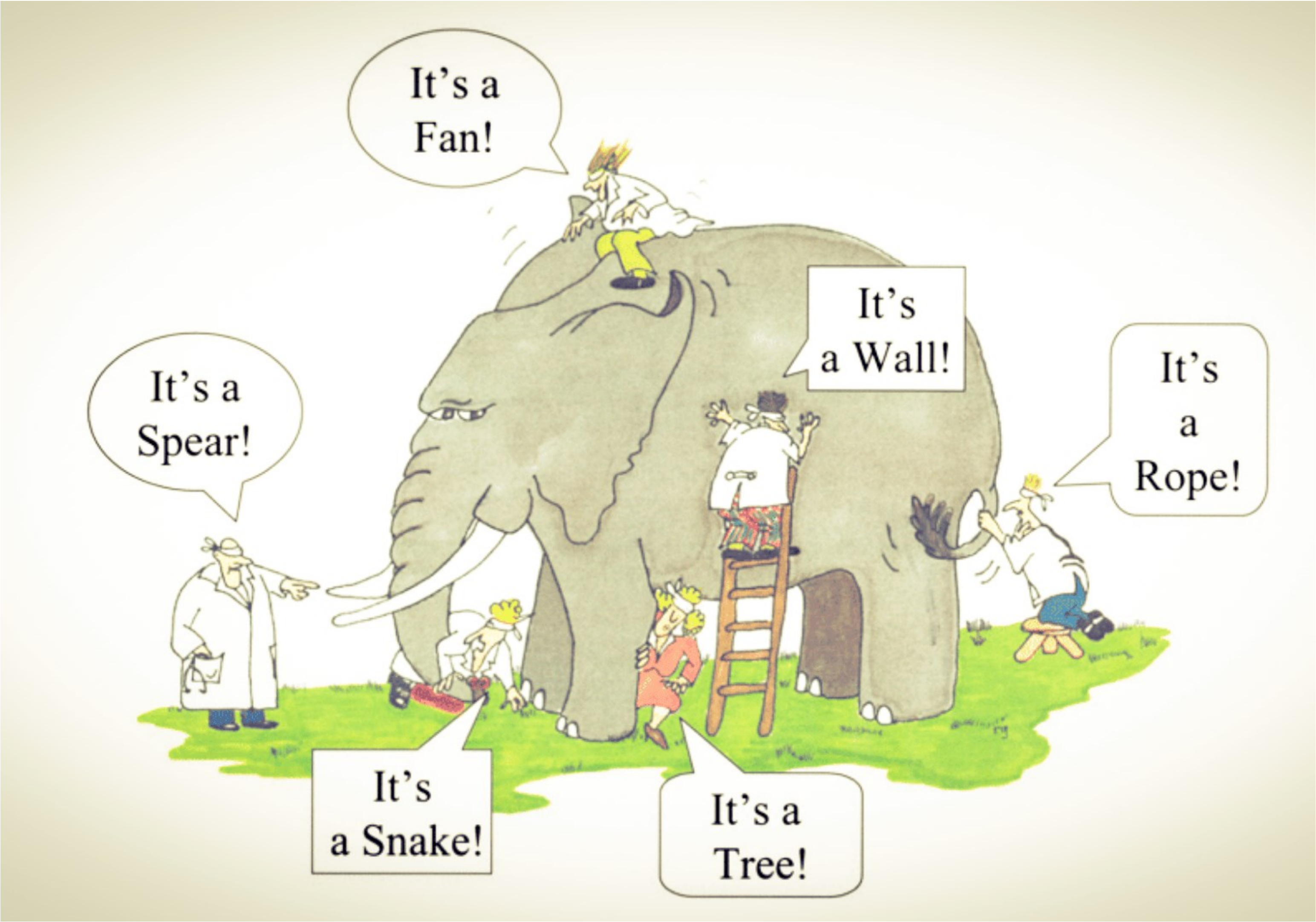
…or, how can we identity a problem to solve?



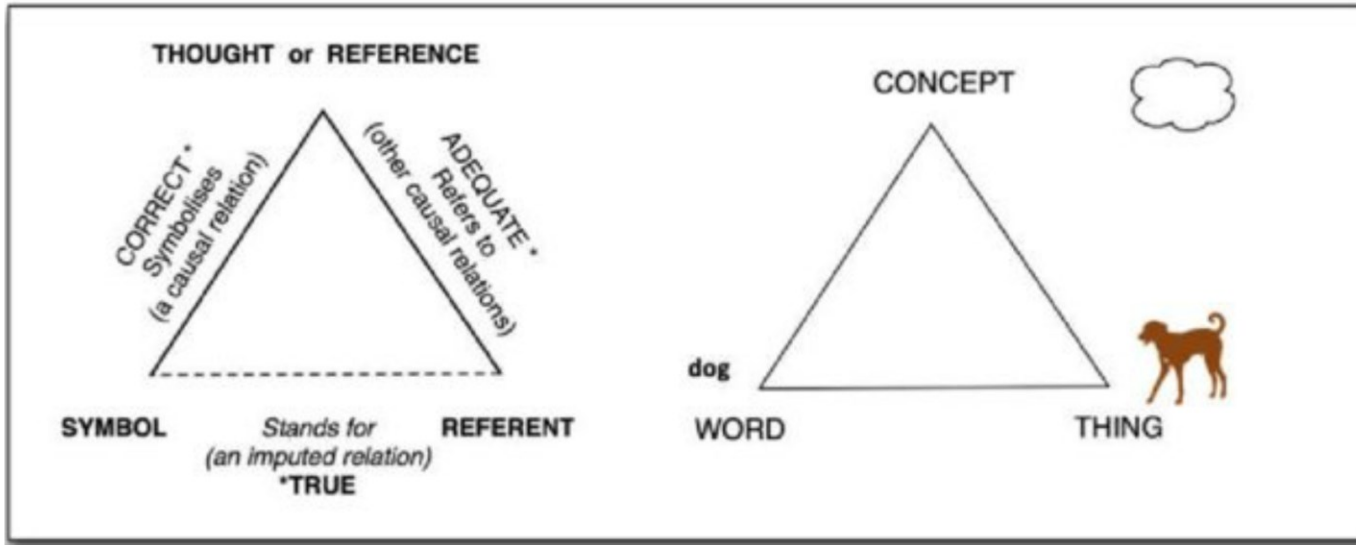
On a broader perspective…

- How to identify a system?
  - How do we become aware of a system?
  - What is shared by commonplace terms such as "solar system", "electrical system", "circulatory system"?
    - What is a "natural" system?
    - What is an "artificial" system?
  - Do we identify or define systems?
- What is an entity, or part (and what is not) of a system?
- Why do we need to model systems?

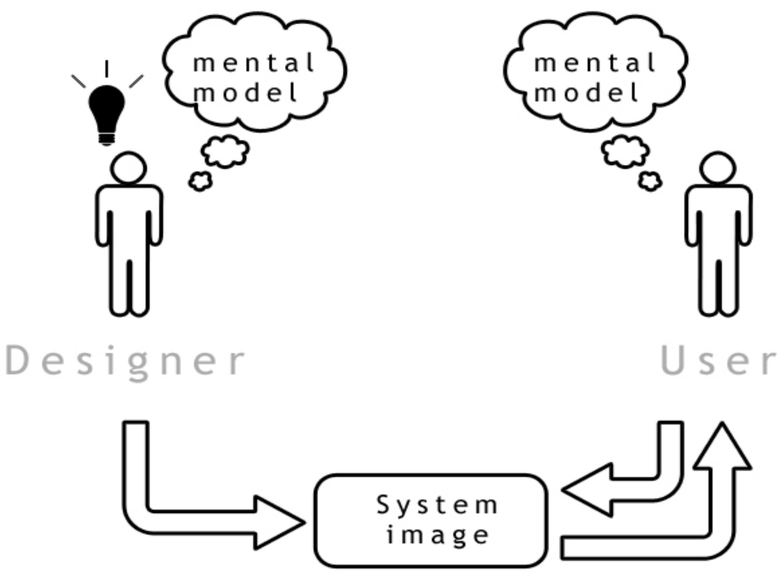
Ok, we are going to address “complicated systems”…
But what for? How? Why?
first…
an Indian fable: “Six blind man and the elephant” (or, how our sensory perceptions can lead to misinterpretations…)



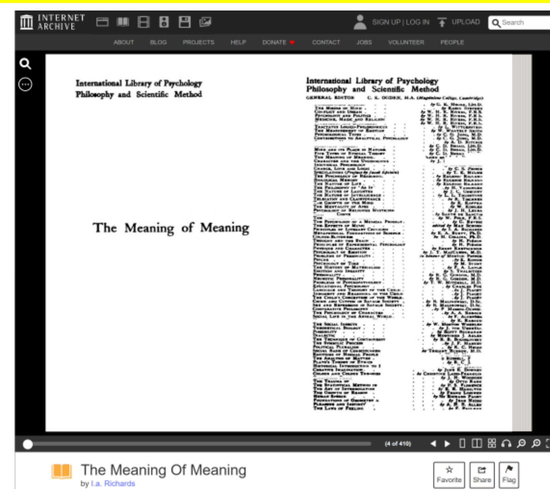
Conceptual models and “The Semiotic Triangle”



See: C. K. Ogden and I. A. Richards (1923) The Meaning of Meaning
Wikipedia: https://en.wikipedia.org/wiki/The_Meaning_of_Meaning
Book online: https://archive.org/details/in.ernet.dli.2015.221615



http://www.interactiondesignblog.com/wp-content/uploads/2008/06/conceptual_models.jpg



Models

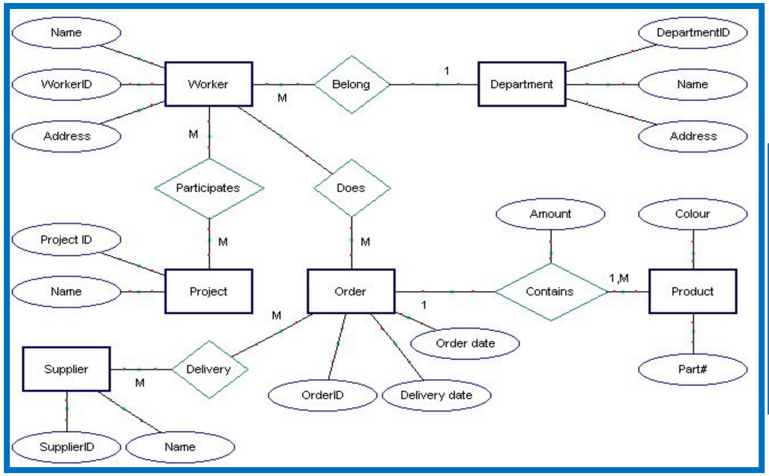
"Because we struggle with the small sizes of our heads as long as we exist, we need intellectual techniques that help us in mastering the complexity we are faced with."

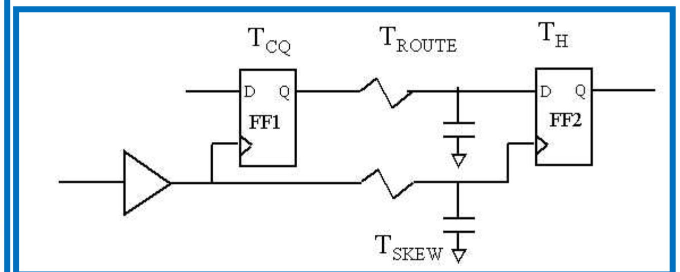
(Edsger W. Dijkstra 1979)

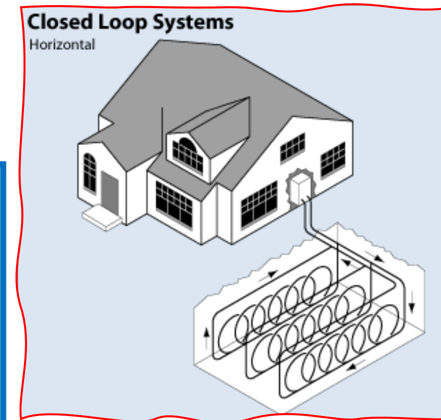
- A model is an **artefact** that represents some**thing**.

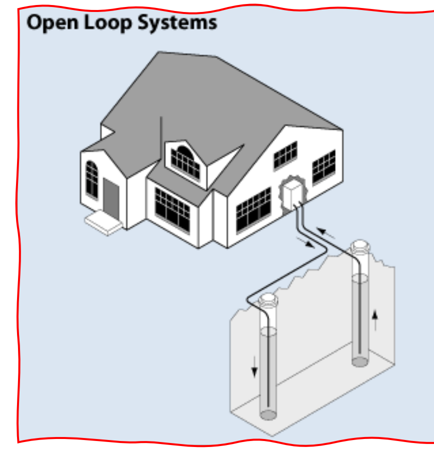
- A model is used to help people to know, to understand and to communicate the subject matter it represent.

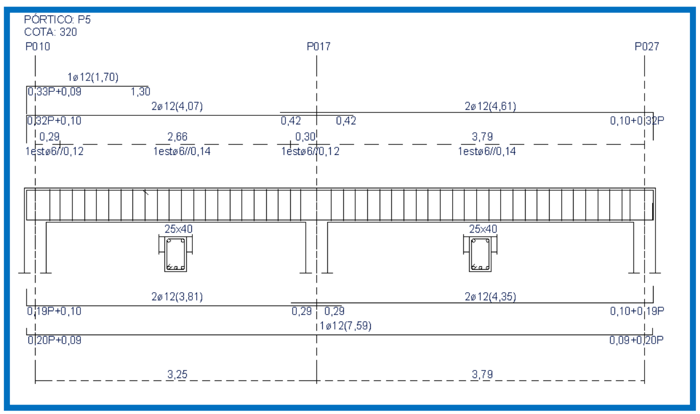
In engineering we can see “models” as a generic technique in use for many purposes… sometimes those models are informal, others they are formal, in the sense that they are made using defined languages…

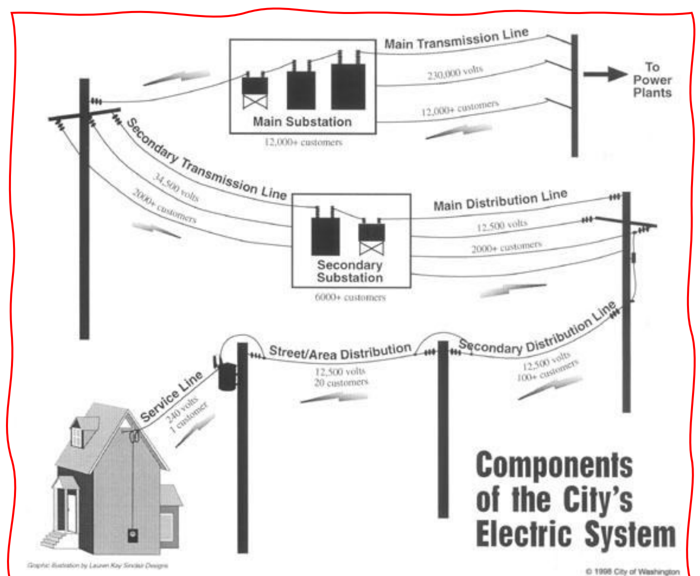


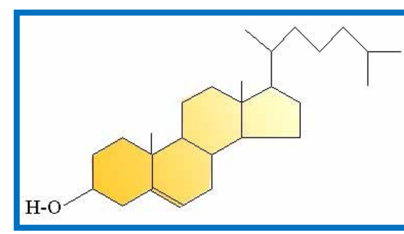












http://www.cartoonstock.com/newscartoons/cartoonists/jco/lowres/jcon1265l.jpg

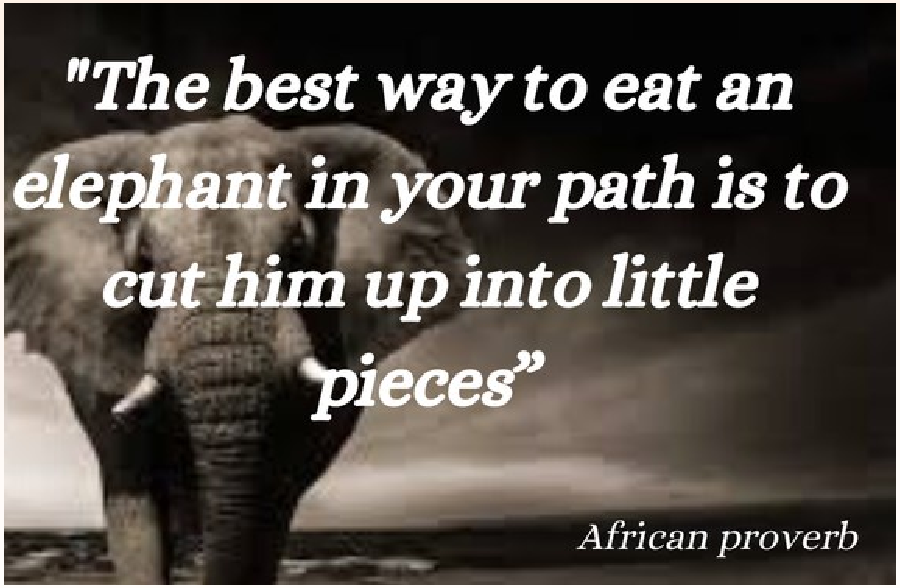
AMS

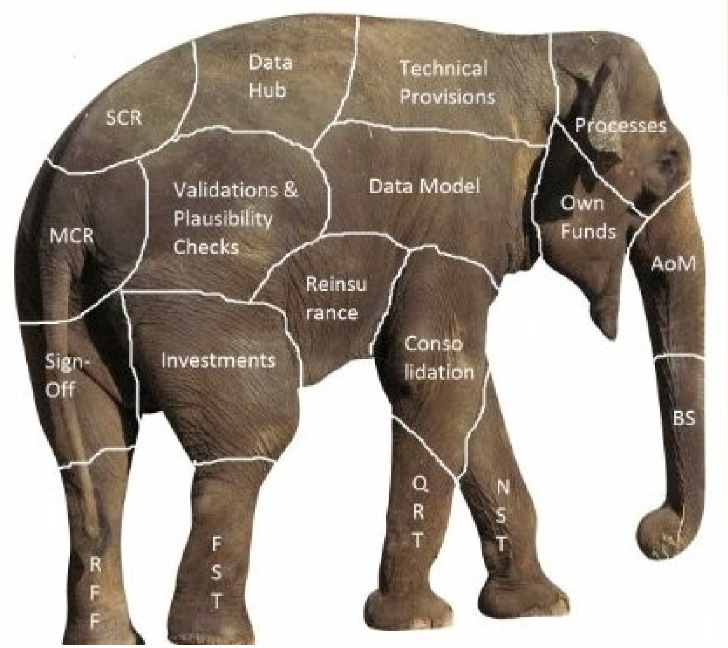
How to make (and make use)

of conceptual models…

Principles, methods and techniques...

Or, in other words, how to make valuable perceptions of the reality…





https://www.jewelry-secrets.com/Blog/do-your-diamond-facets-line-up/

Conceptual Models

- A **conceptual model** is a nonphysical model that describes a (real world) state of affairs.

- There are many types of conceptual models, e.g.:
  - Philosophical models (e.g. teleological, ontological, epistemological, metaphysical).
  - Scientific models (e.g. logical, mathematical, statistical).
  - Social and political models (e.g. macro-economic, economic, socio-technical).
  - Information system models (e.g. soft-systems, entity-relationship, domain).

Always observe the 2 principles:

- **100 Percent principle:**
  - All relevant general static and dynamic aspects, *i.e.*, all rules, laws, *etc.*, of the universe of discourse should be described in the conceptual schema. The information system cannot be held responsible for not meeting those described elsewhere, including in particular those in application programs.

- **Conceptualization Principle:**
  - A conceptual schema should only include conceptually relevant aspects, both static and dynamic, of the universe of discourse, thus excluding all aspects of (external or internal) data representation, physical data organization and access, as well as all aspects of particular external user representation such as message formats, data structures, etc.

The Orange Report ISO TR9007, https://www.brcommunity.com/articles.php?id=b479

How to evaluate if the model is fit for purpose?

- **Adequacy for purpose**
  - adequacy$_i$: A Model (M) is *adequate$_i$-for-P* if and only if using M in instance I **results in the achievement** of purpose P.
  - adequacy$_c$: A Model (M) is *adequate$_c$-for-P* if and only if, in Context-of-Use-type instances of use of M, purpose P is **very likely** to be achieved.

- **Fitness for purpose**
  - fit$_x$: A Model (M) is *fit$_x$-for-P* if and only if it is *adequate$_x$-for-P*.
  - fitness$_x$: A Model (M)'s *fitness$_x$-for-P* is greater to the extent that M is *adequate$_x$-for-P* for a higher-ranking member of $P=\{P_{min},\ldots,P_{max}\}$.

Model Evaluation: An Adequacy-for-Purpose View,
https://www.cambridge.org/core/journals/philosophy-of-science/article/model-evaluation-an-adequacyforpurpose-view/CA91669E7CAC8BE4332A2B6D99BC9DB0

Purpose of a model

- **"An adequate model is one that is sufficient for the purposes of interest not just as a matter of accident (*e.g.*, a one-off accurate prediction) but because the model has properties that make it suitable for those purposes."**

Parker, Wendy S. 2011. "When Climate Models Agree: The Significance of Robust Model Predictions." Philosophy of Science 78 (4): 579–600

- **"An adequate model is one that has properties that promote the kind of model output which is desired."**

Currie, Adrian. 2018. "From Models-as-Fictions to Models-as-Tools." Ergo 4 (27): 759–81

RoME = Return on Modelling Effort

http://erikproper.blogspot.com/2009/02/models-that-matter-return-on-modelling.html

- Measures the returned benefit by the creation, or the improvement, of a model considering the applied effort (similar with ROI for investments)

- Serves as a stop criteria to the modelling process!

RoME = Return on Modelling Effort

“stop criteria” to the modelling process => objectives of modeling:

- Understand (let me/us try to understand the problem...)
- Communicate (...let us talk about the problem and find solutions...)
- Build (...let me/us build/change/maintain the solutions...)
- Automate (...let me/us use the solution...)

...

“Bases de dados”

“***Análise e Modelação de Sistemas***”

“Integração Empresarial”

“Arquitetura Empresarial”

“Gestão de Projetos Informáticos”

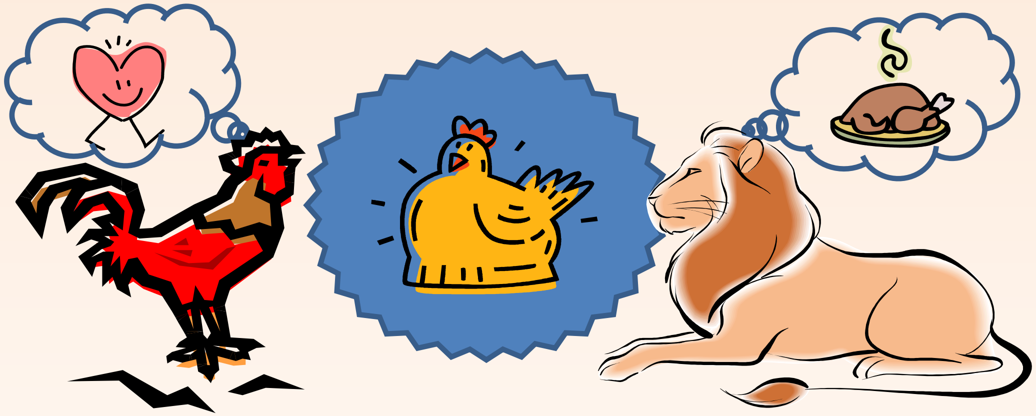
“Engenharia e Tecnologia dos Processos de Negócio”

...

AMS

On Models and Languages !!!

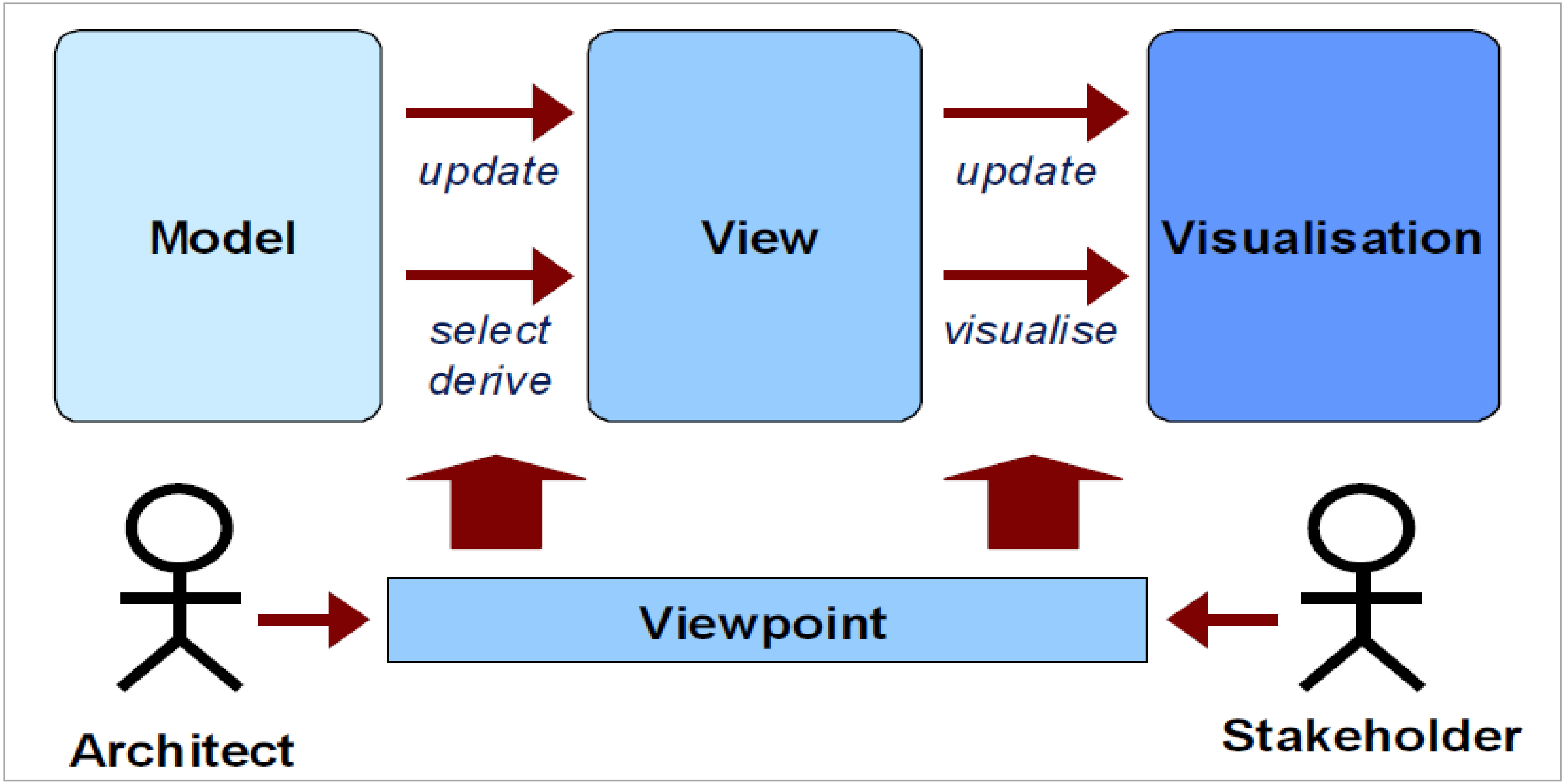
Or...



On Views, Viewpoints, and Models, …

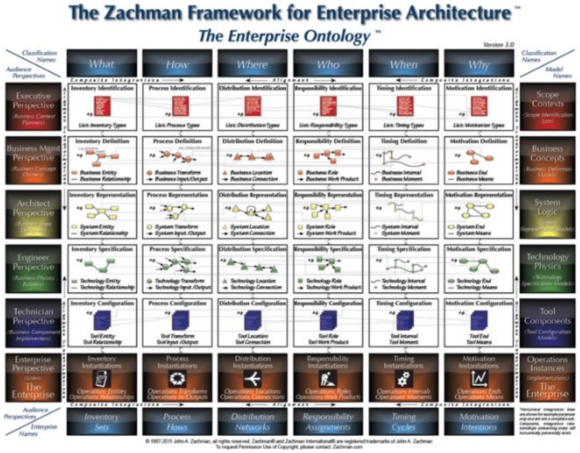
…also “Black box” versus “White box”

Architecture, Model, Views, Viewpoints



Architecture defines the fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution .

“viewpoints” and “views”…

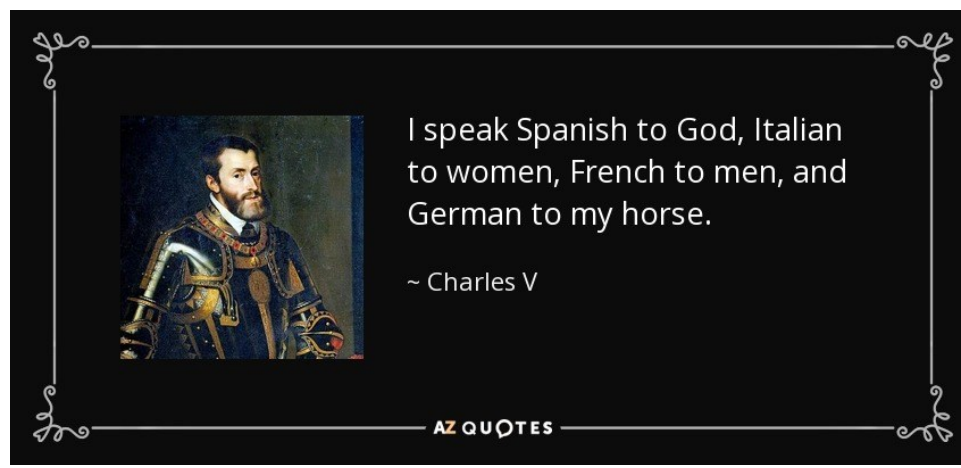


	Why	How	What	Who	Where	When
Contextual	Goal List	Process List	Material List	Organisational Unit & Role List	Geographical Locations List	Event List
Conceptual	Goal Relationship	Process Model	Entity Relationship Model	Organisational Unit & Role Relationship Model	Locations Model	Event Model
Logical	Rules Diagram	Process Diagram	Data Model Diagram	Role Relationship Diagram	Locations Diagram	Event Diagram
Physical	Rules Specification	Process Function Specification	Data Entity Specification	Role Specification	Location Specification	Event Specification
Detailed	Rules Details	Process Details	Data Details	Role Details	Location Details	Event Details

https://www.zachman.com/images/ZI_PIcs/ZF3.0.jpg
https://en.wikipedia.org/wiki/Zachman_Framework#/media/File:The_Zachman_Framework_of_Enterprise_Architecture.jpg

Challenge: how to "talk" about all that?

	Why	How	What	Who	Where	When
Contextual	Goal List	Process List	Material List	Organisational Unit & Role List	Geographical Locations List	Event List
Conceptual	Goal Relationship	Process Model	Entity Relationship Model	Organisational Unit & Role Relationship Model	Locations Model	Event Model
Logical	Rules Diagram	Process Diagram	Data Model Diagram	Role Relationship Diagram	Locations Diagram	Event Diagram
Physical	Rules Specification	Process Function Specification	Data Entity Specification	Role Specification	Location Specification	Event Specification
Detailed	Rules Details	Process Details	Data Details	Role Details	Location Details	Event Details



https://www.azquotes.com/quote/1307063

https://bigguystudio.ca/babel

Walid Saba is a Senior Research Scientist at the Institute for Experiential AI at Northeastern University. Prior to joining the institute in 2023, he worked at two Silicon Valley startups, focusing on conversational AI. This work included high-level roles as the principal AI scientist for telecommunications company Astound and CTO of software company Klangoo, where he helped develop its state-of-the-art digital content semantic engine (Magnet). Saba's career to date has seen him hold various positions in both the private sector and academia. His resume includes entities such as the American Institutes for Research, AT&T Bell Labs, IBM and Cognos, while he has also spent a cumulative seven years teaching computer science at the University of Ottawa, the New Jersey Institute of Technology (NJIT), the University of Windsor (a public research university in Ontario, Canada), and the American University of Beirut (AUB). He has published over 45 technical articles, including an award-winning paper that he presented at the German Artificial Intelligence Conference (KI-2008). Walid received his BSc and MSc in Computer Science from the University of Windsor, and a Ph.D in Computer Science from Carleton University in 1999.

evaluating linguistic competency
interpretation is a mapping

	to **formal**	to **informal**
from **formal**	*Java to Python SQL to SPARQL json to XML RDF to OWL etc.*	*Express information in relational structures (KG or RDB) in natural language*
from **informal**	*Convert unstructured content into formal structures (KG or RDB) ——— Answer NL questions posed to a relational structure (RDB or KG)*	*Search Summarization Sentiment Analysis Topic Modeling Data Extraction Translation etc.*

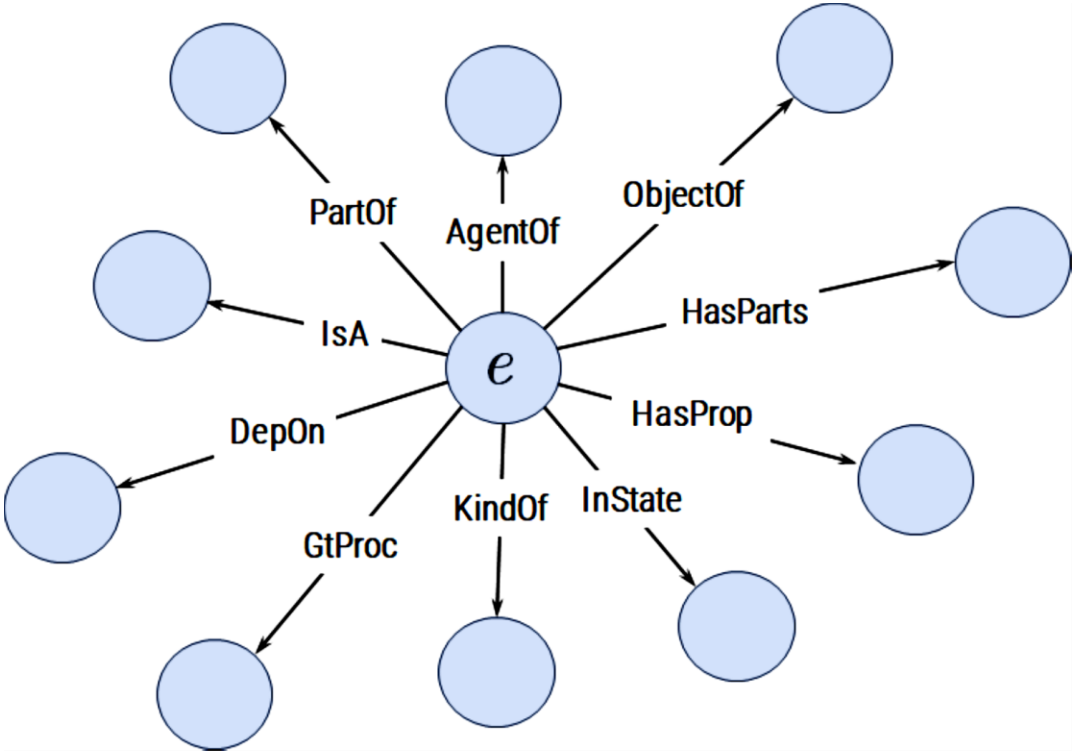
- not relevant to language understanding
- good test of language understanding
- does not test language understanding

Currently most benchmarks are based on evaluating LLMs on these subjective tasks that cannot be objectively evaluated

21

Walid Saba is a Senior Research Scientist at the Institute for Experiential AI at Northeastern University. Prior to joining the institute in 2023, he worked at two Silicon Valley startups, focusing on conversational AI. This work included high-level roles as the principal AI scientist for telecommunications company Astound and CTO of software company Klangoo, where he helped develop its state-of-the-art digital content semantic engine (Magnet). Saba's career to date has seen him hold various positions in both the private sector and academia. His resume includes entities such as the American Institutes for Research, AT&T Bell Labs, IBM and Cognos, while he has also spent a cumulative seven years teaching computer science at the University of Ottawa, the New Jersey Institute of Technology (NJIT), the University of Windsor (a public research university in Ontario, Canada), and the American University of Beirut (AUB). He has published over 45 technical articles, including an award-winning paper that he presented at the German Artificial Intelligence Conference (KI-2008). Walid received his BSc and MSc in Computer Science from the University of Windsor, and a Ph.D in Computer Science from Carleton University in 1999.

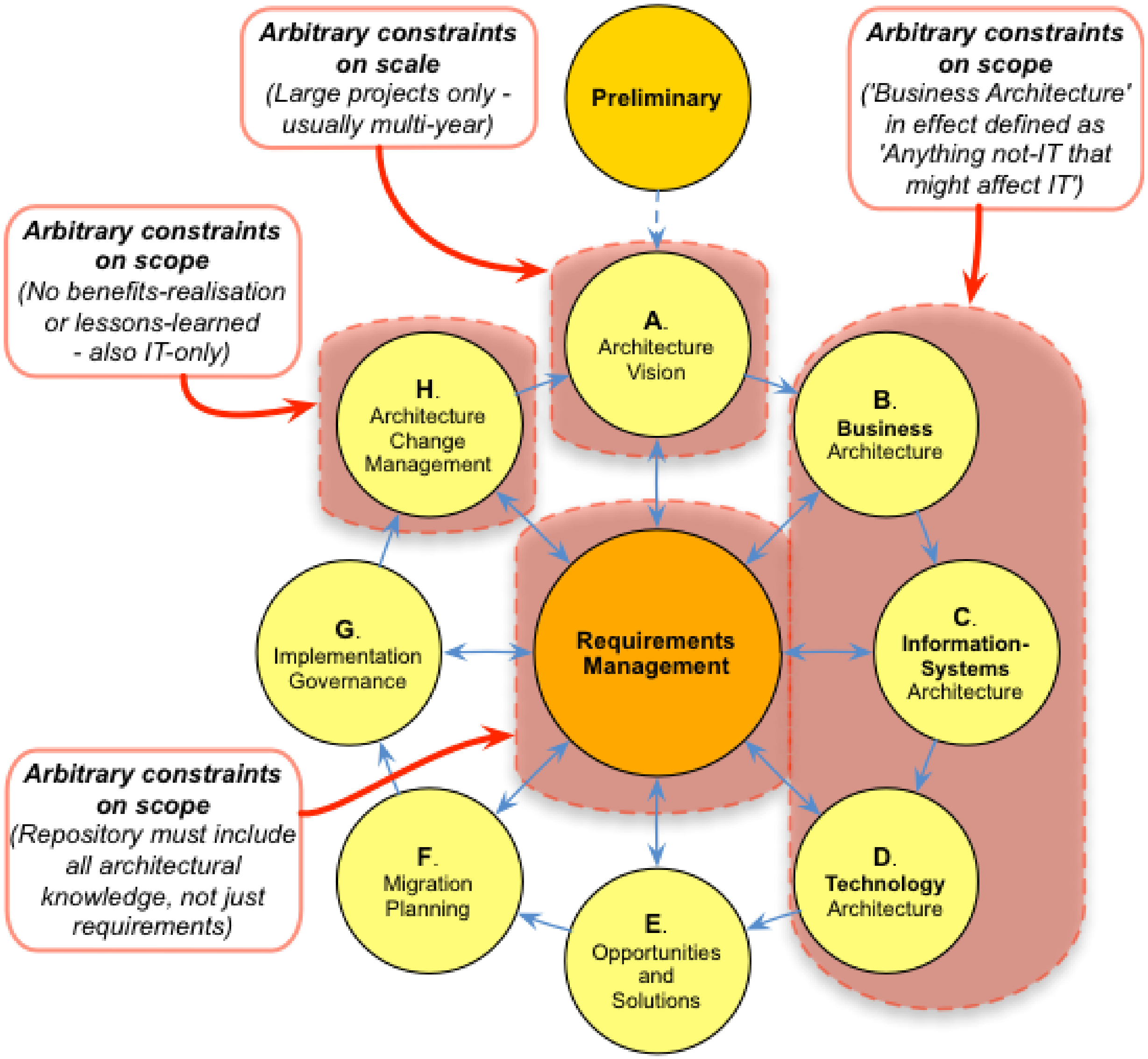
discovering the dimensions of meaning



33

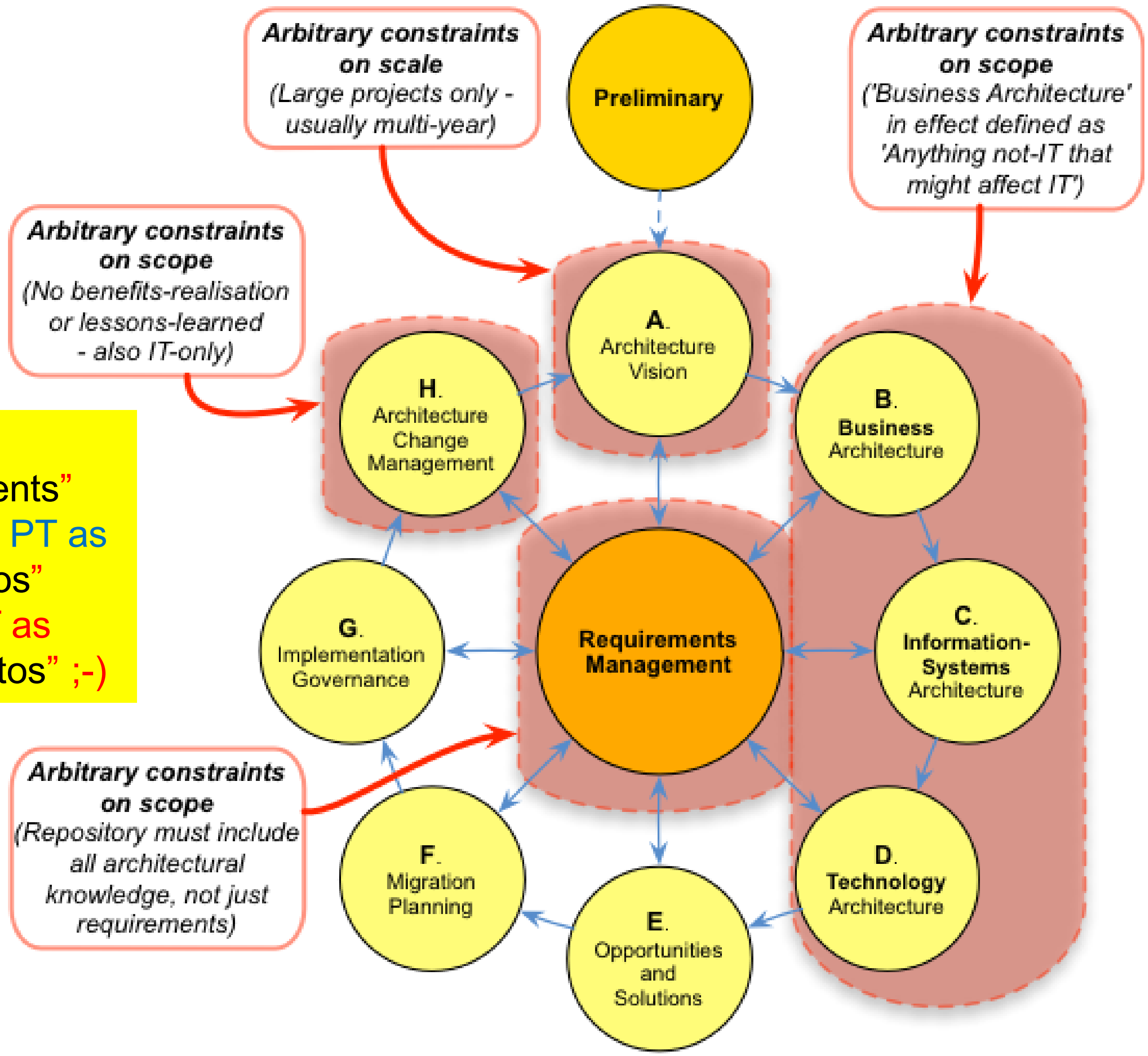
Methods

When the system is… a business… An Enterprise Architecture method: ADM (from the TOGAF framework…)



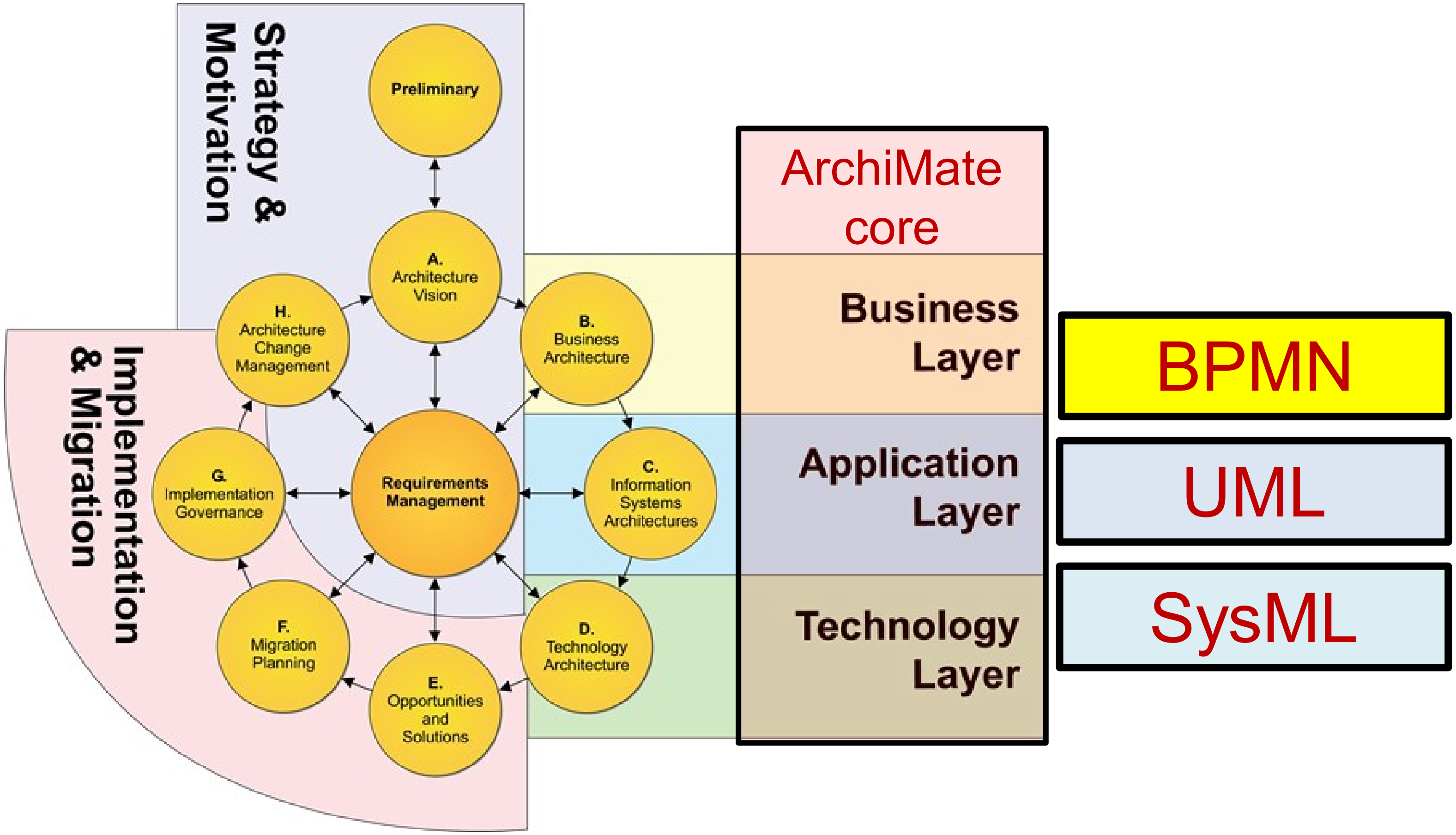
Source: http://weblog.tetradian.com/wp-content/uploads/2014/11/togaf-side-by-side.png

…in other words, it will be all about "requirements management" and "architecture"



BTW, "requirements" translates to PT as "requisitos" and NOT as "requerimentos" ;-)

We can do a lot having ADM as reference, but for now we'll focus here on only a part of it, with "ArchiMate" (core), "BPMN", "UML" and "SysML"…



Source: http://www.opengroup.org/sites/default/files/contentimages/Certification/Archimate/documents/archimate-tog-mapping-small_0.jpg

ArchiMate - high level modelling of a business and its technology

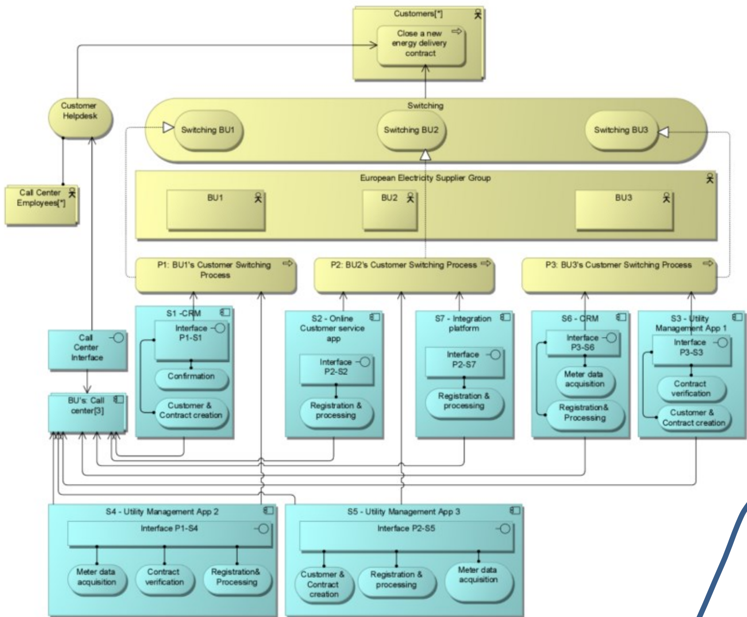
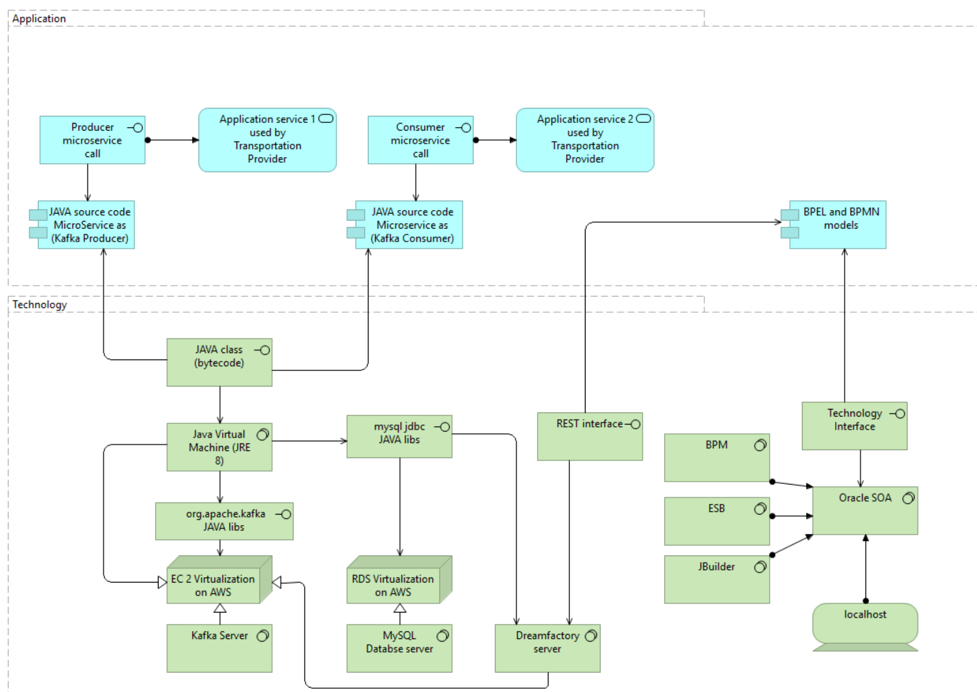
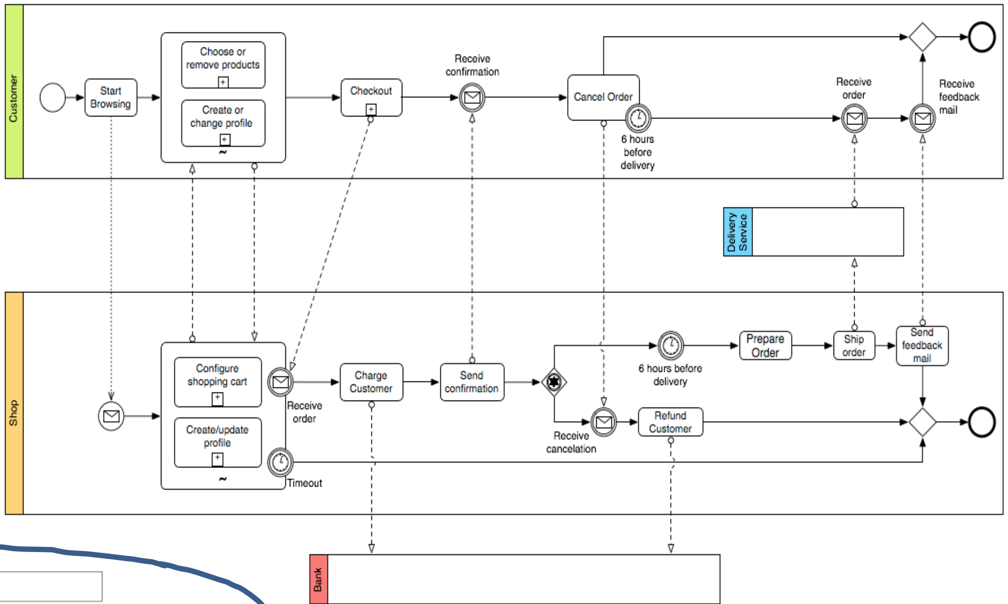


Figure 15 - Baseline architecture



Process Engineering… with BPMN, a formal language!!!



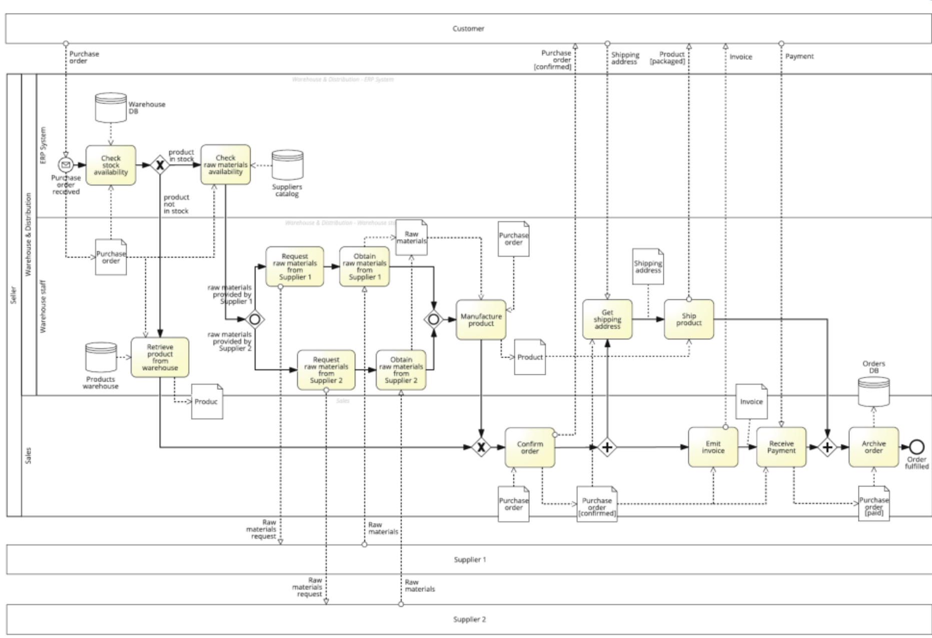


Fig. 3.16 Collaboration diagram between a seller, a customer, and two suppliers

UML - modelling applications and data lifecycles

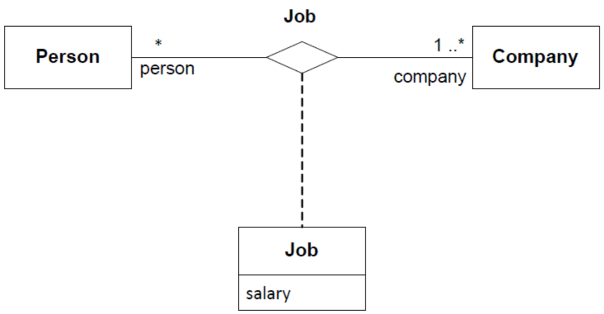
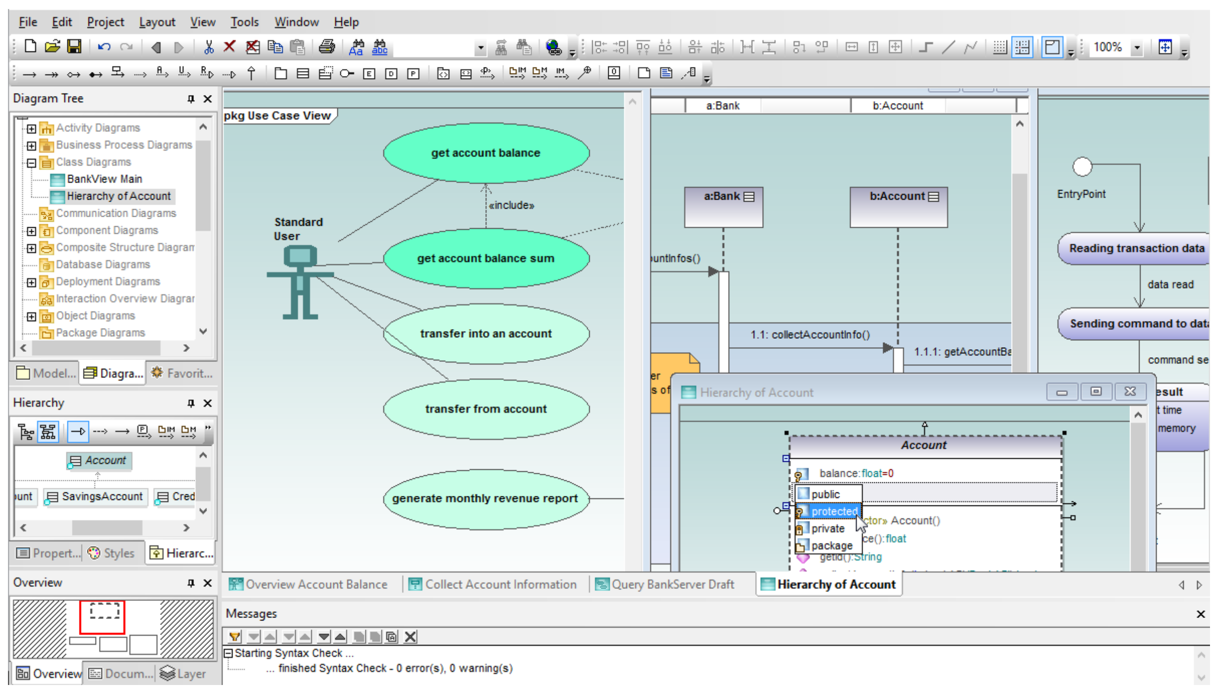
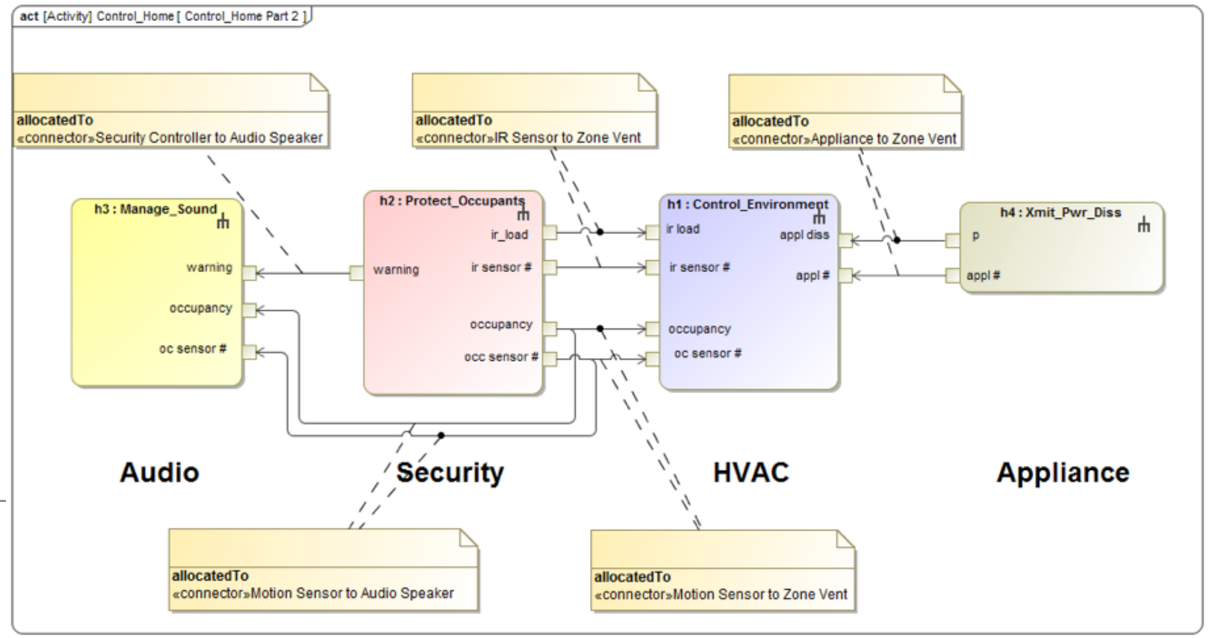
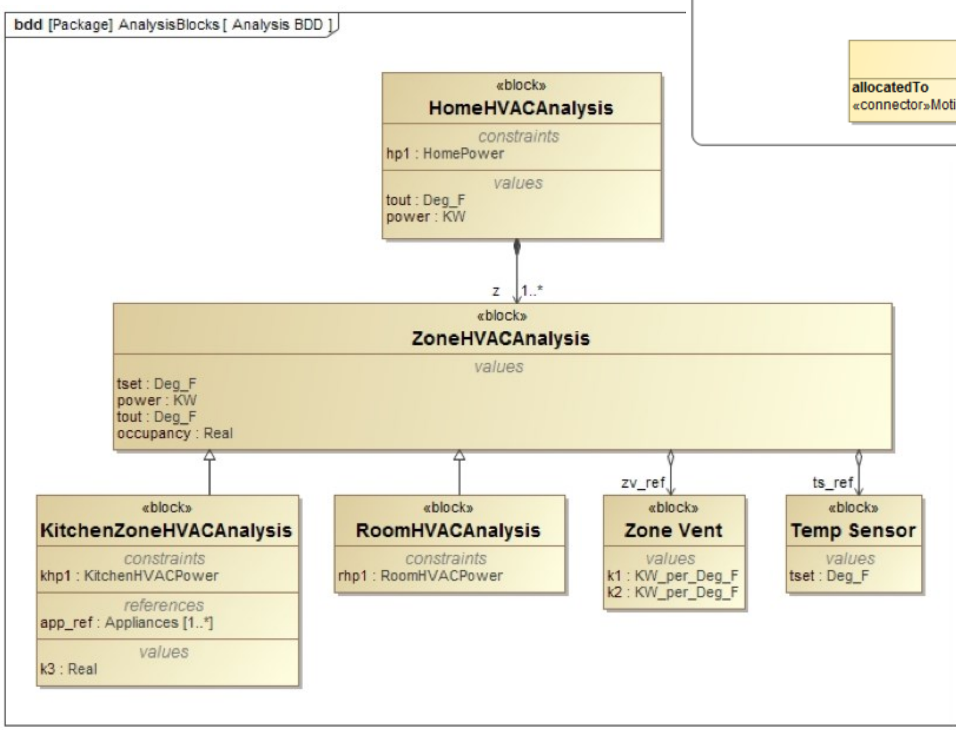


Figure 11.36 Example AssociationClass using diamond symbol



SysML - modelling of cyber-physical concerns in systems…





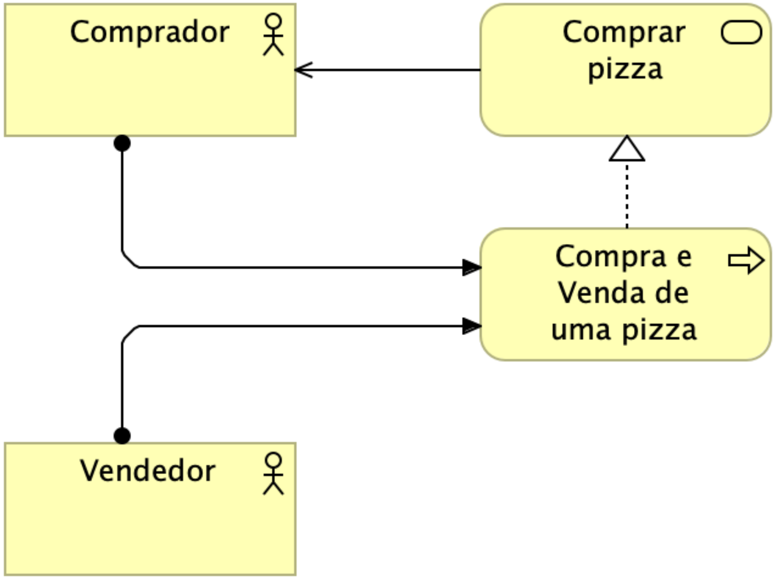
https://intercax.com/2015/12/20/smart-home-iot-2/

Hello World! ArchiMate

Ao **processo de negócio** de compra e venda de uma pizza **estão assignados** dois **atores de negócio**: o comprador e o vendedor.

O **serviço de negócio** comprar pizza é **realizado** por este mesmo **processo de negócio** de compra e venda de uma pizza.

Por sua vez, o **serviço de negócio** comprar pizza **serve** o comprador.

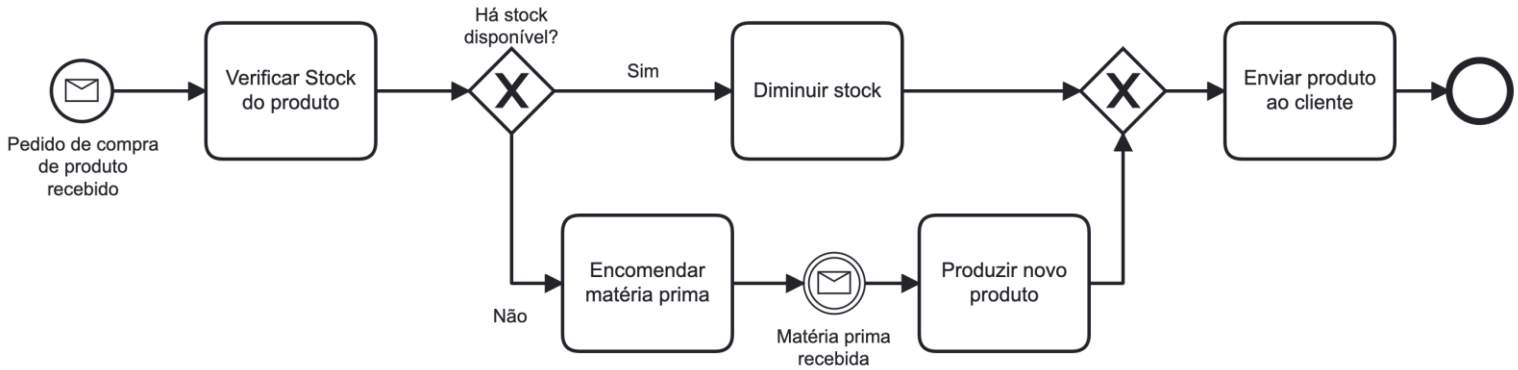


Hello World! BPMN

Quando um vendedor recebe um pedido de compra de produto verifica sempre o stock disponível.

Se houver stock disponível, atualiza-o e envia o produto ao cliente.

Se não houver stock disponível, encomenda matéria-prima e espera pela sua receção. De seguida, produz o novo produto e envia-o ao cliente.



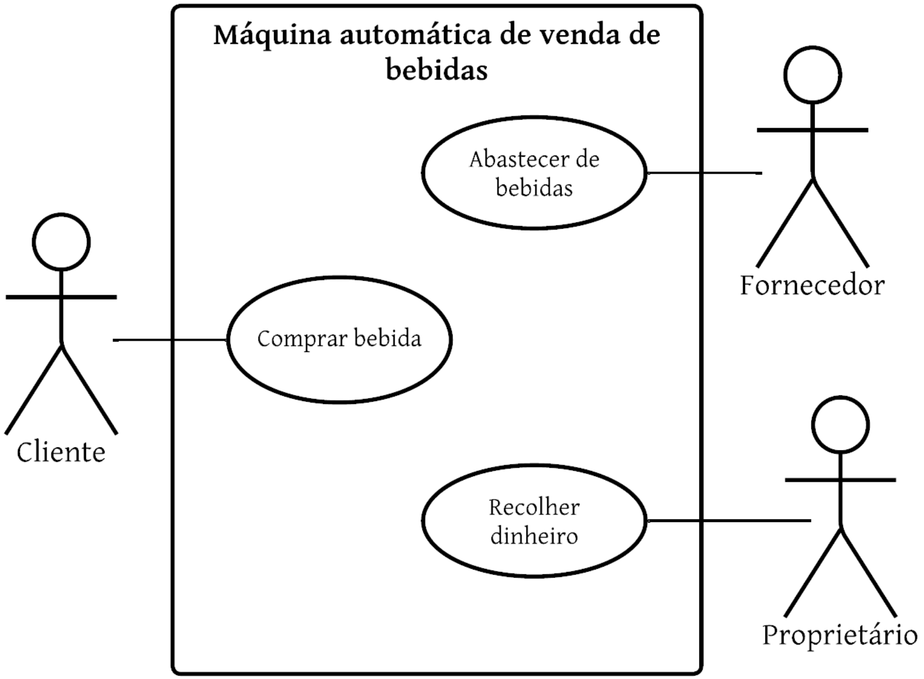
Hello World! UML - UC

Uma máquina automática de venda de bebidas tem as seguintes capacidades:

A compra de bebida é feita diretamente pelo cliente.

O abastecimento de stock de bebidas é feito pelo fornecedor.

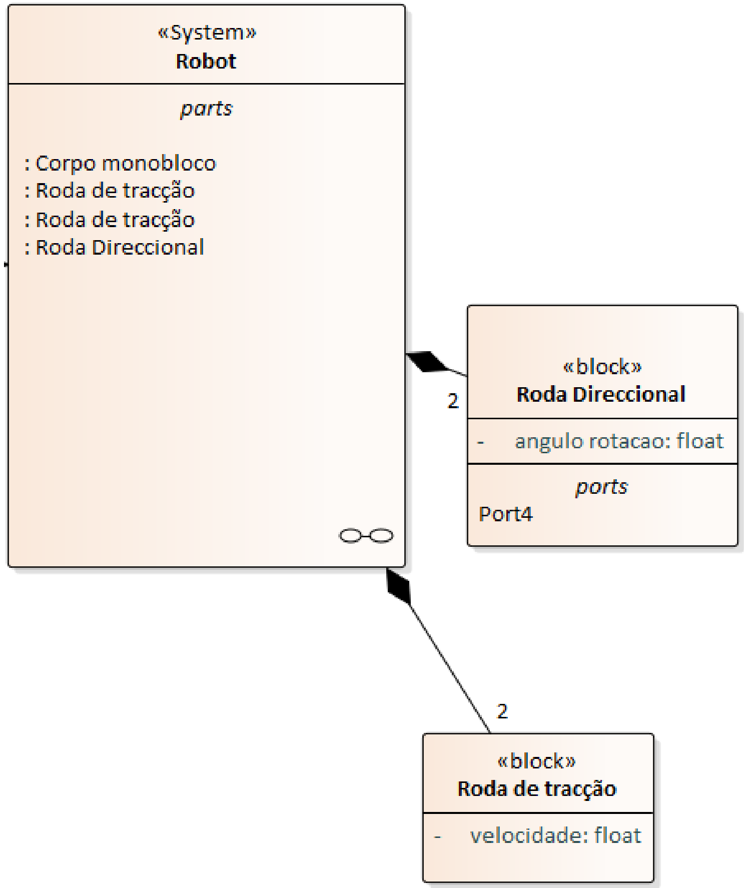
O proprietário da máquina de bebidas recolhe o dinheiro da venda de bebidas.



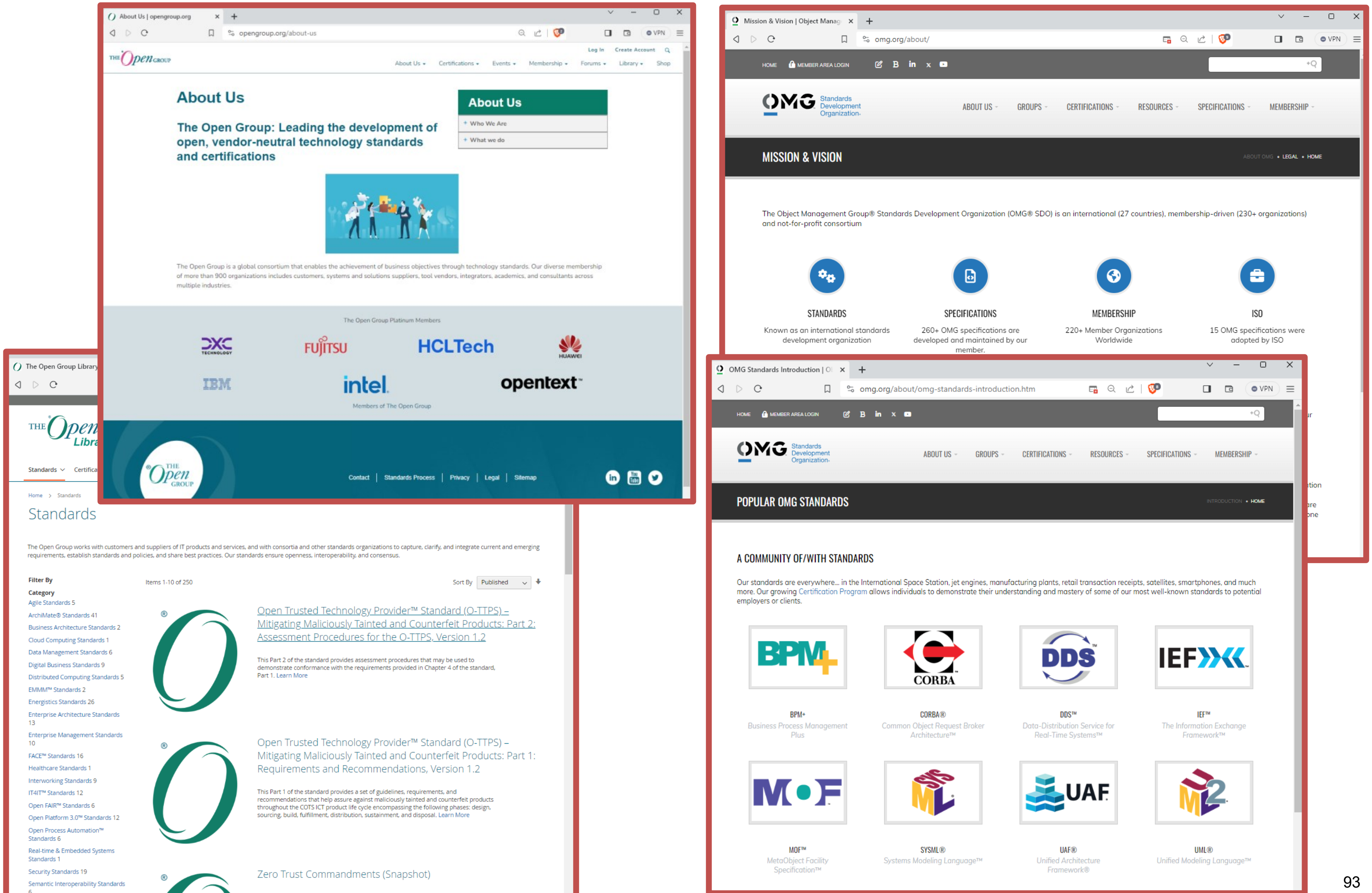
Hello World! SysML

Um robot de vigilância é constituído pelos seguintes componentes físicos:

- 2 rodas direcionais com ângulo de rotação definido em graus
- 2 rodas de tração que permitem a sua deslocação
- 1 corpo monobloco



Who is behind these languages and methods?
“The Open Group” and the “Object Management Group”…

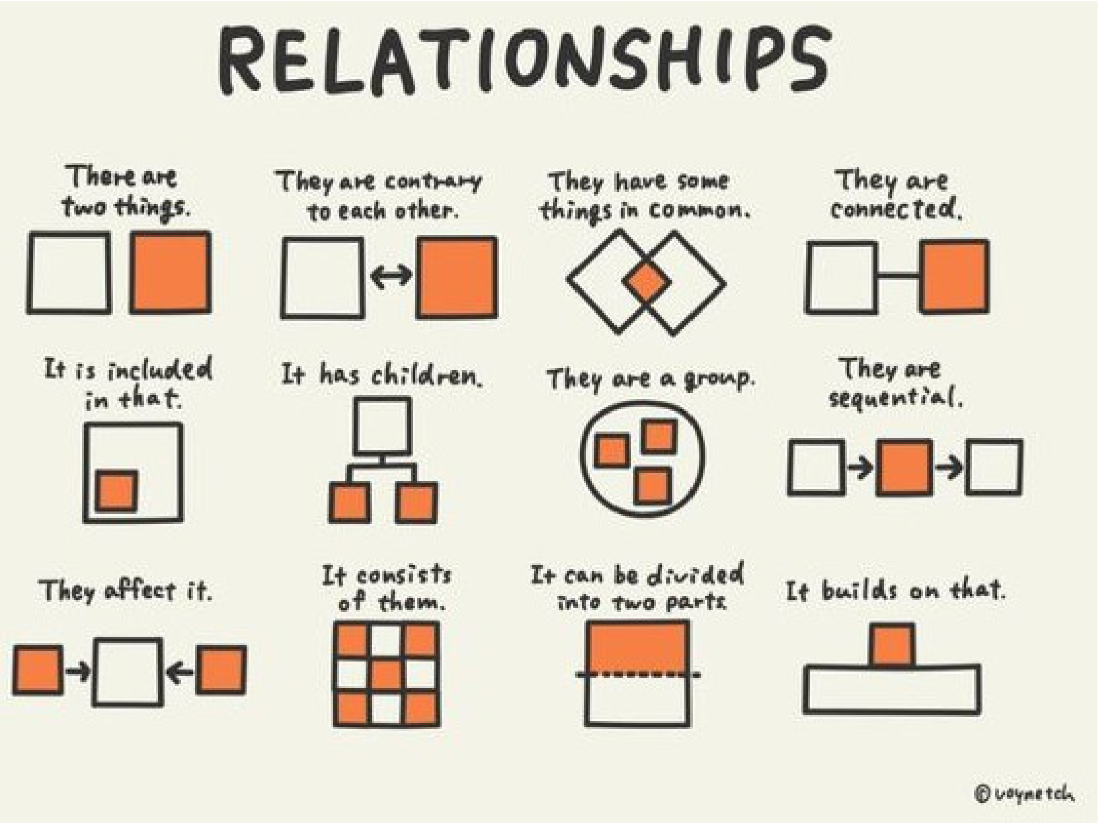


Structure & Behaviour

Structure and behaviour are two general ways of classifying **viewpoints**.

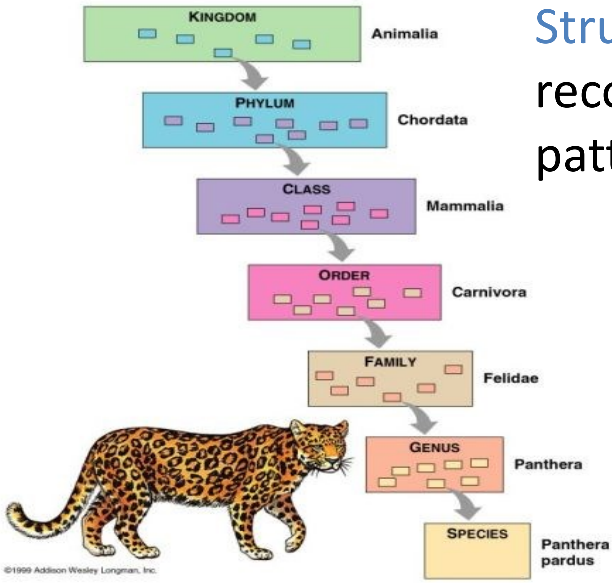
Ok, but are they so relevant to us?

Because we assume that structure and behavior are usually related to concerns that most of the time can pragmatically be addressed separated. ("**orthogonal separation of concerns**")

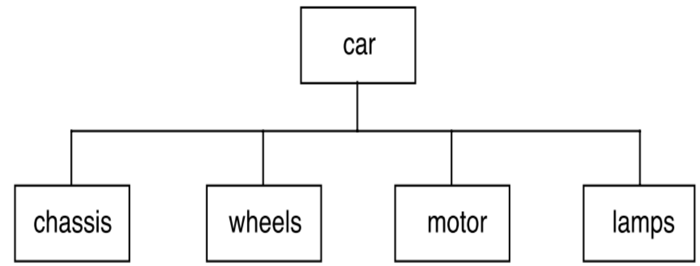


https://www.pinterest.jp/voynetch/visual-note-taking/

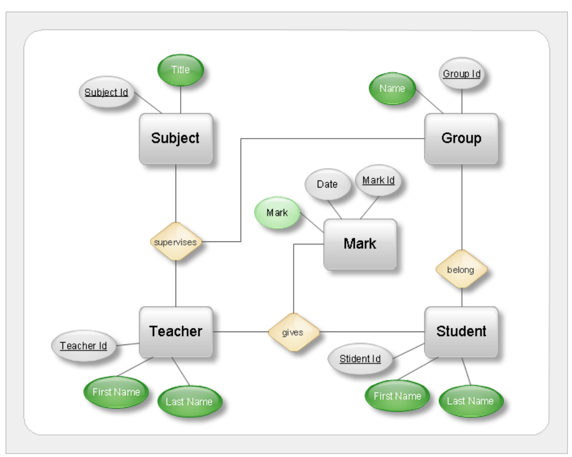
Structure

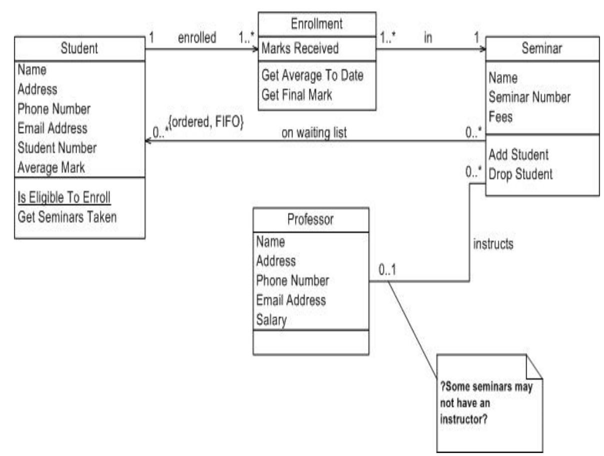


Structure is a fundamental notion referring to the recognition, observation, nature, and stability of patterns and relationships of entities.

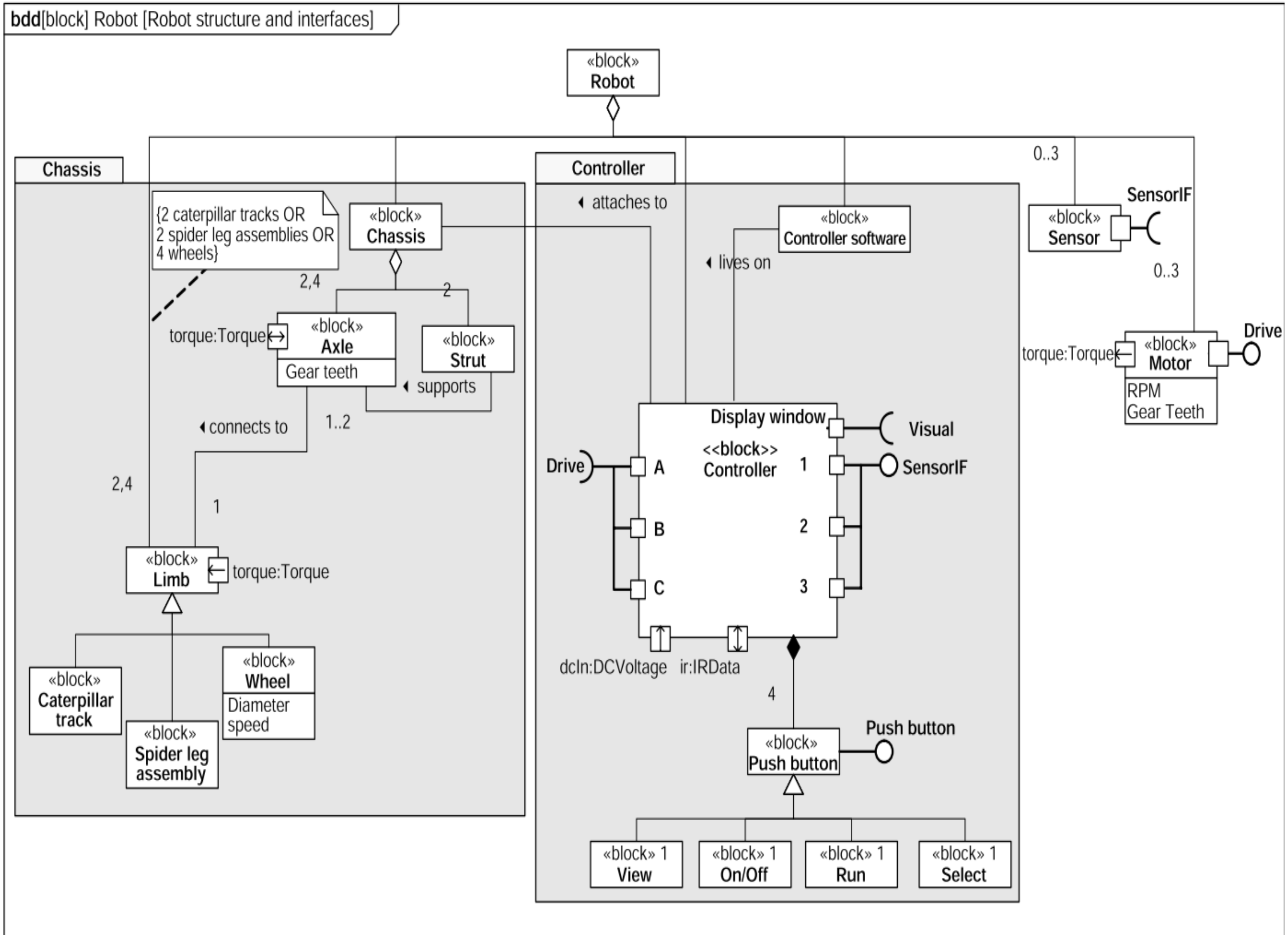


Structural representations of “logical things”:





btw, now in SysML, a conceptual model of a “physical thing”, a robot ;-)

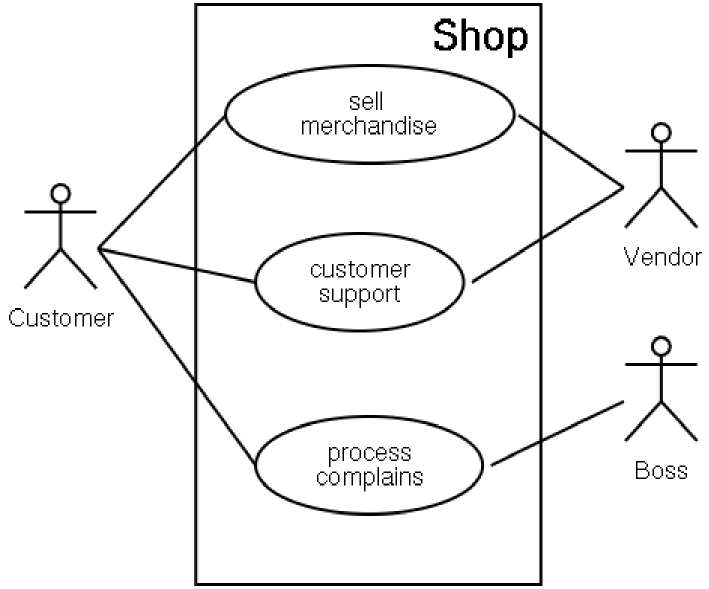


Behavior

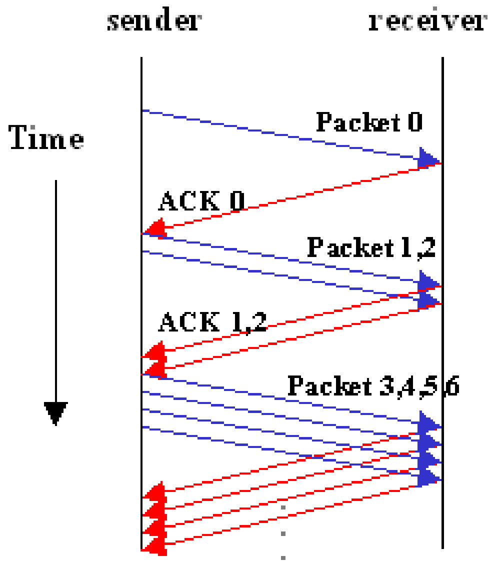
Behavior refers to the actions of a system.

It is the response of the system to various stimuli (inputs), whether external (triggered by system's actors) or internal (triggered by changes in the state of the system, due to its memory capability and sequences of causalities).

An example of a use case diagram (UC)

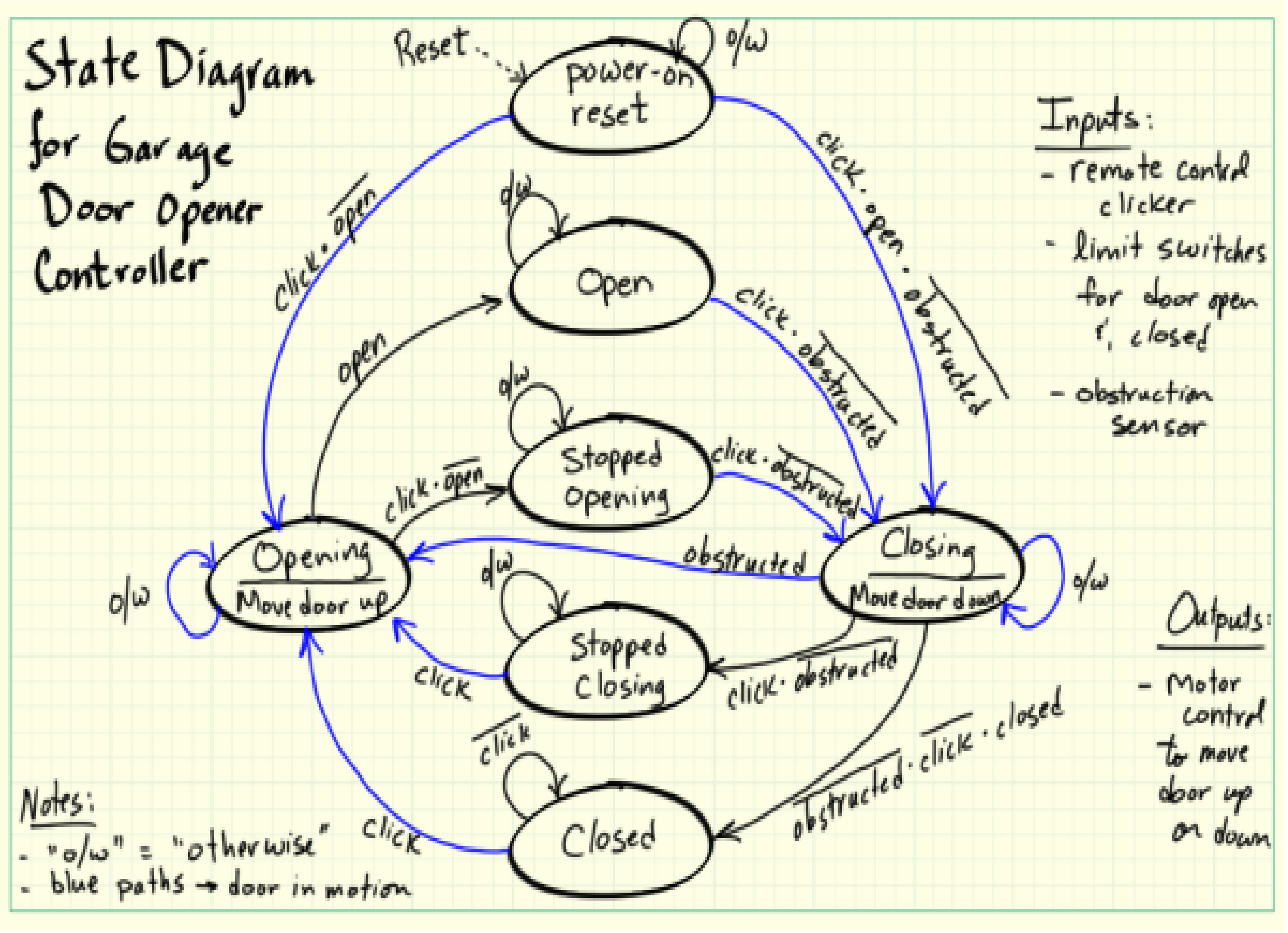


An example of a sequence diagram (SEQ)



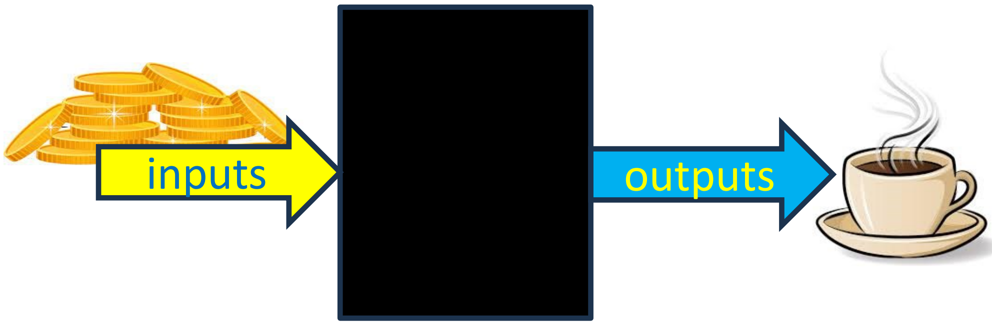
Behavior...

An example of a state machine diagram (STM)



About “black-box” versus “white-box”
(quoting http://bulldozer00.com/2012/07/25/extrapolation-abstraction-modeling/)

Black-box views: focus in
the value of the system for its actors



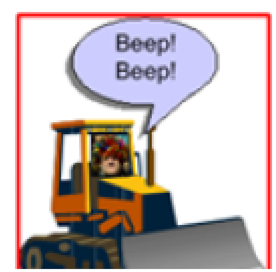
White-box views: focus in
the internal structure and behavior of the system

*Note that in Bill’s model, there is an **internal** goal that determines the response to a given “disturbance“. Thus, given the same disturbance at two different points in time, the white box model can generate different responses whereas the black box model would always generate the same response.*
For example, the white box model explains anomalies like why, on the 100th test run, a mouse won’t press a button to get a food pellet as it did on the 99 previous runs. In this case, the internal goal may be to “eat until satiated“. When the internal goal is achieved, the externally observed behavior changes because the stimulus is no longer important to the mouse.
Theories based on extrapolation and abstraction are useful for predicting short term actions and trends within a certain probability, but when a physical model of the underlying mechanisms of a phenomenon is discovered, it explains a lot of anomalies unaccounted for by extrapolation/abstraction.
For a taste of Mr. Powers’ control system-based theory of behavior, download and experiment with the software provided here: Living Control Systems III.

Thanks Bulldozeroo

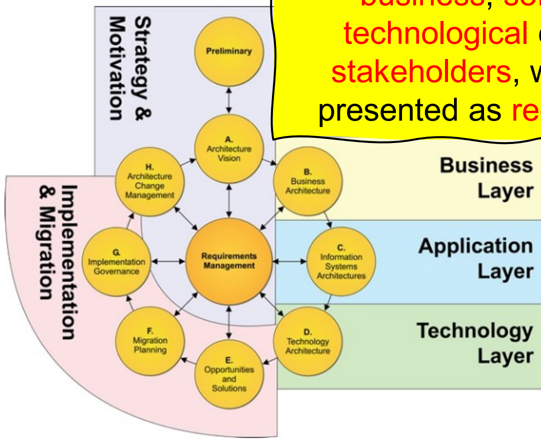
(http://bulldozeroo.com/)!

Interesting “food for tought”...



Finally, this is what will be our focus here:

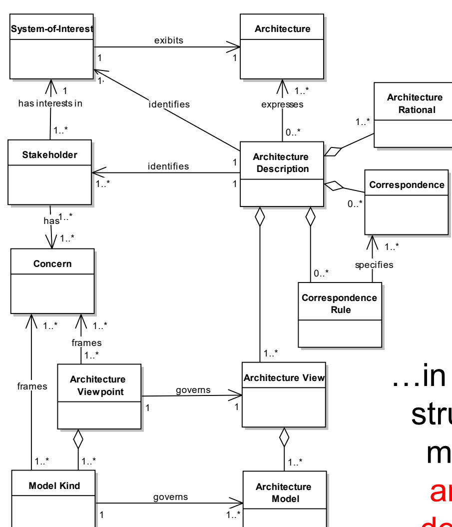
…how to capture the main business, software and technological concerns of stakeholders, which can be presented as requirements…



…to express a shareable understanding of a system of concern…

	Why	How	What	Who	Where	When
Contextual	Goal List	Process List	Material List	Organisational Unit & Role List	Geographical Locations List	Event List
Conceptual	Goal Relationship	Process Model	Entity Relationship Model	Organisational Unit & Role Relationship Model	Locations Model	Event Model
Logical	Rules Diagram	Process Diagram	Data Model Diagram	Role Relationship Diagram	Locations Diagram	Event Diagram
Physical	Rules Specification	Process Function Specification	Data Entity Specification	Role Specification	Location Specification	Event Specification
Detailed	Rules Details	Process Details	Data Details	Role Details	Location Details	Event Details

…in the form of a structured and manageable architecture description…



…that can be valuable to inform during the life-cycle of that system…

SO, what is the purpose of Modelling?

Put simply, to help people (the *stakeholders*)

to communicate in unambiguous ways

in order

to solve problems

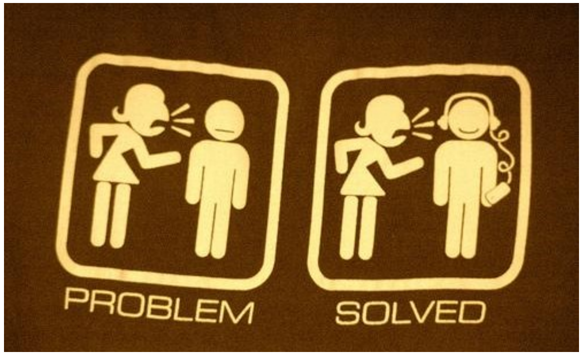
(which, btw, is exactly what engineers are expected to do…)

OK, but what is therefore the difficulty of that?

The difficulty is that:

If language is not correct, then what is said is not what is meant.

(Confucius)



http://pictures.funnyjunksite.com/wp-content/uploads/2010/03/Funny-T-shirt-Designs-6.jpg