From Systems Engineering to Software Engineering: A Perspective for Systems-of-Systems

Prof. Dr. Elisa Yumi Nakagawa
Prof. Dr. José Carlos Maldonado

SSC5964: Software Engineering ||1st semester, 2016

Agenda

- System Engineering
- Software Engineering
- Systems-of-Systems
- Conclusions

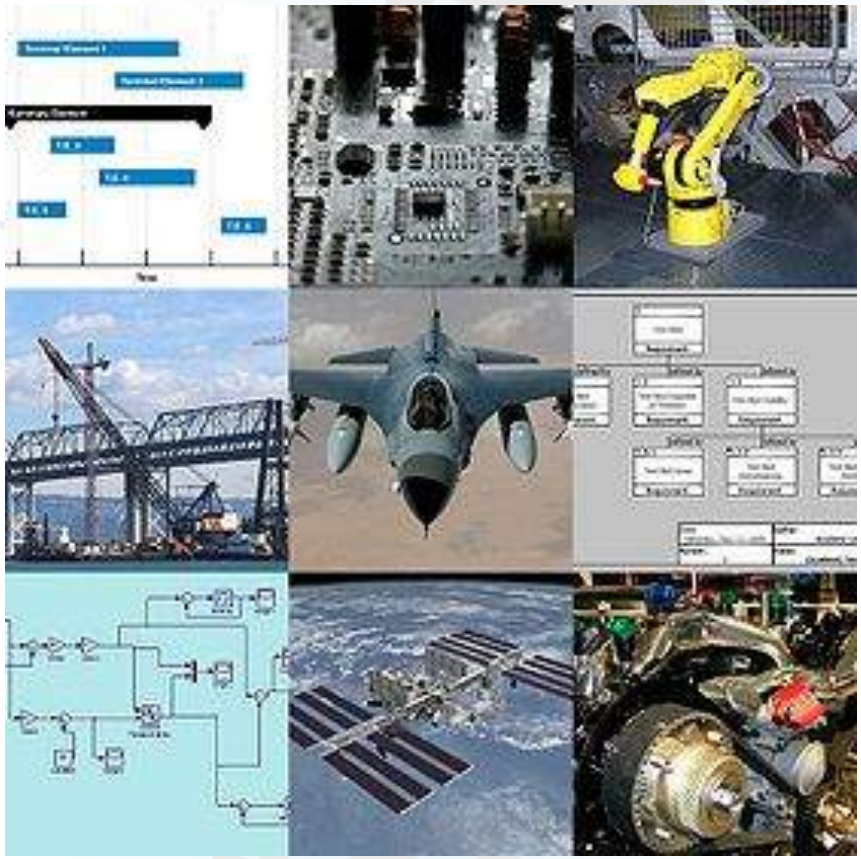
System Engineering

Definition

- Systems engineering
  - is an interdisciplinary field of engineering that focuses on how to design and manage complex engineering systems (with regard to time, budget, and other constraints).
  - Term coined in the 1940s
- INCOSE (International Council on Systems Engineering)
  - From 1990 (NCOSE), 1995 (INCOSE)
  - http://www.incose.org

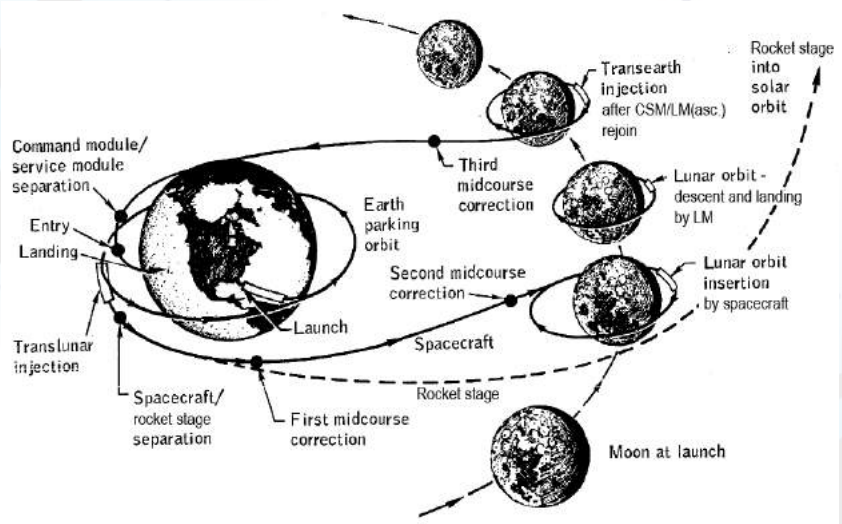
Systems

- Spacecrafts
- Computer chip
- Robotics
- Bridges
- Metropolis/Cities
- Buildings
- Automobiles
- Others

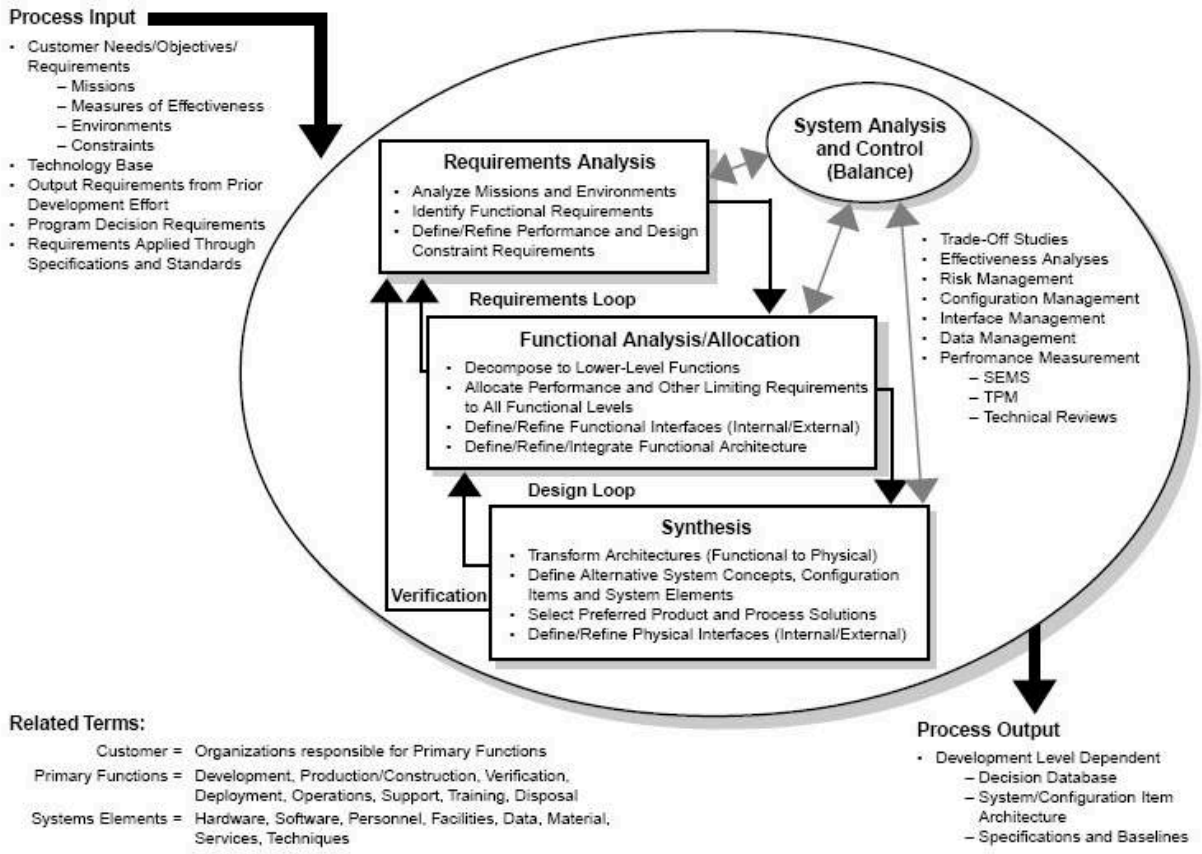


Example

- Apollo program
  - A leading example of a systems engineering project
  - First humans on the Moon (1969-1972)
  - Interdisciplinarity
    - Physics, Mechanical/Electrical Engineering, Medical areas, Project Management,…, Computation (Hardware/Software)



System Engineering Process



Systems Engineering Fundamentals. Defense Acquisition University Press, 2001 (http://www.dau.mil/publications/publicationsDocs/SEFGuide%2001-01.pdf)

Software Engineering

From Origin to Now

- Origin:
  - "The dissemination of knowledge is of obvious value — the massive dissemination of error-loaded software is frightening." [*Dijkstra*]
  - http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF

- Software Engineering (SE):
  - "The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software." [IEEE Standard Glossary of Software Engineering Terminology," IEEE std 610.12-1990]
  - Sub-field of Engineering
  - Part of overall Systems Engineering

From Origin to Now

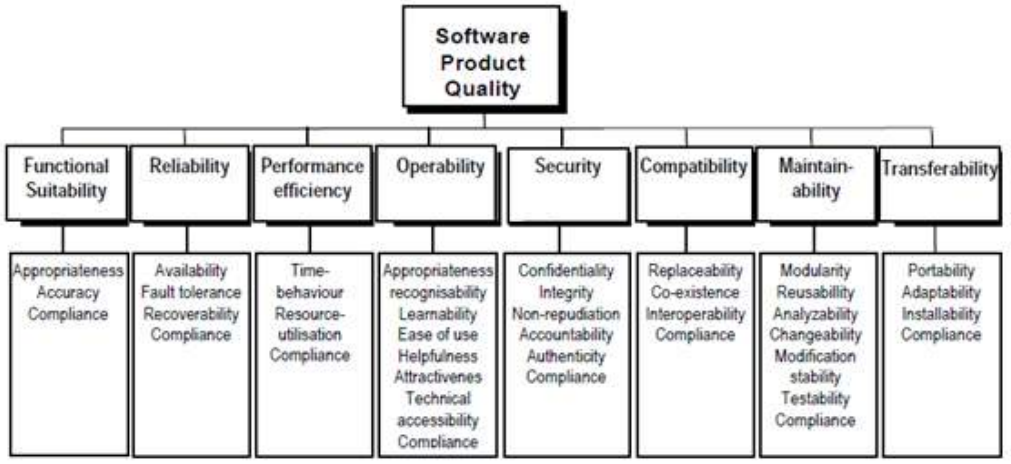
- Today
  - Conferences, workshops, and journals/magazine
    - E.g., ICSE, SBES, WICSA, JSS, ACM TOSEM, IEEETSE, IEEESw, ESE, SE Notes
  - SE research groups
  - SE researchers/practitioners
  - Supporting tools
    - Commercial, open source, and academic
  - Software companies
    - Big companies (e.g., IBM, Google, SAP), many other SME

From Origin to Now

- Today
  - Hundreds of books
    - PRESSMAN, R. S.; MAXIM, B.; Software Engineering: An Practitioner's Approach, McGraw-Hill Education, 8th Edition, 2014.
    - SOMMERVILLE, I.; Software Engineering, Pearson, 9th Edition, 2010.
    - PFLEEGER, S. L.; ATLEE, J. M.; Software Engineering: Theory and Practice, Prentice Hall, 4th Edition, 2009.
    - GHEZZI, C.; JAZAYERI, M.; MANDRIOLI, D.; Fundamentals of Software Engineering. Pearson, 2nd Edition, 2002.
    - VON MAYRHAUSER, A.; Software Engineering: Methods and Management, Academic Press, 1990.
    - …

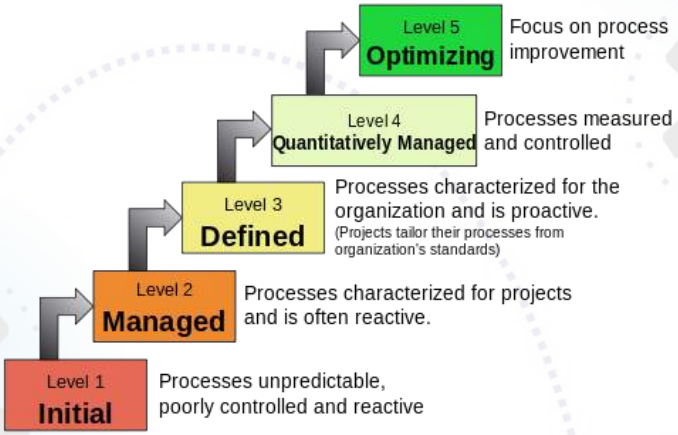
SE: Main Goal

- Main goal
  - product/process quality improvement
- Process quality ➔ Software product quality
- Software product quality
  - ISO/IEC 9126 and ISO/IEC 25010
  - Specific quality models (e.g., SoS and CES)



Main Goal: Quality

- Process quality
  - CMMI (http://cmmiinstitute.com)



  - ISO/IEC 15504 Information technology – Process assessment
    - ISO/IEC 33001:2015 Information technology – Process assessment – Concepts and terminology, March/2015

SE Overview

- ISO/IEC 12207-2008: Systems and software engineering -Software life cycle processes

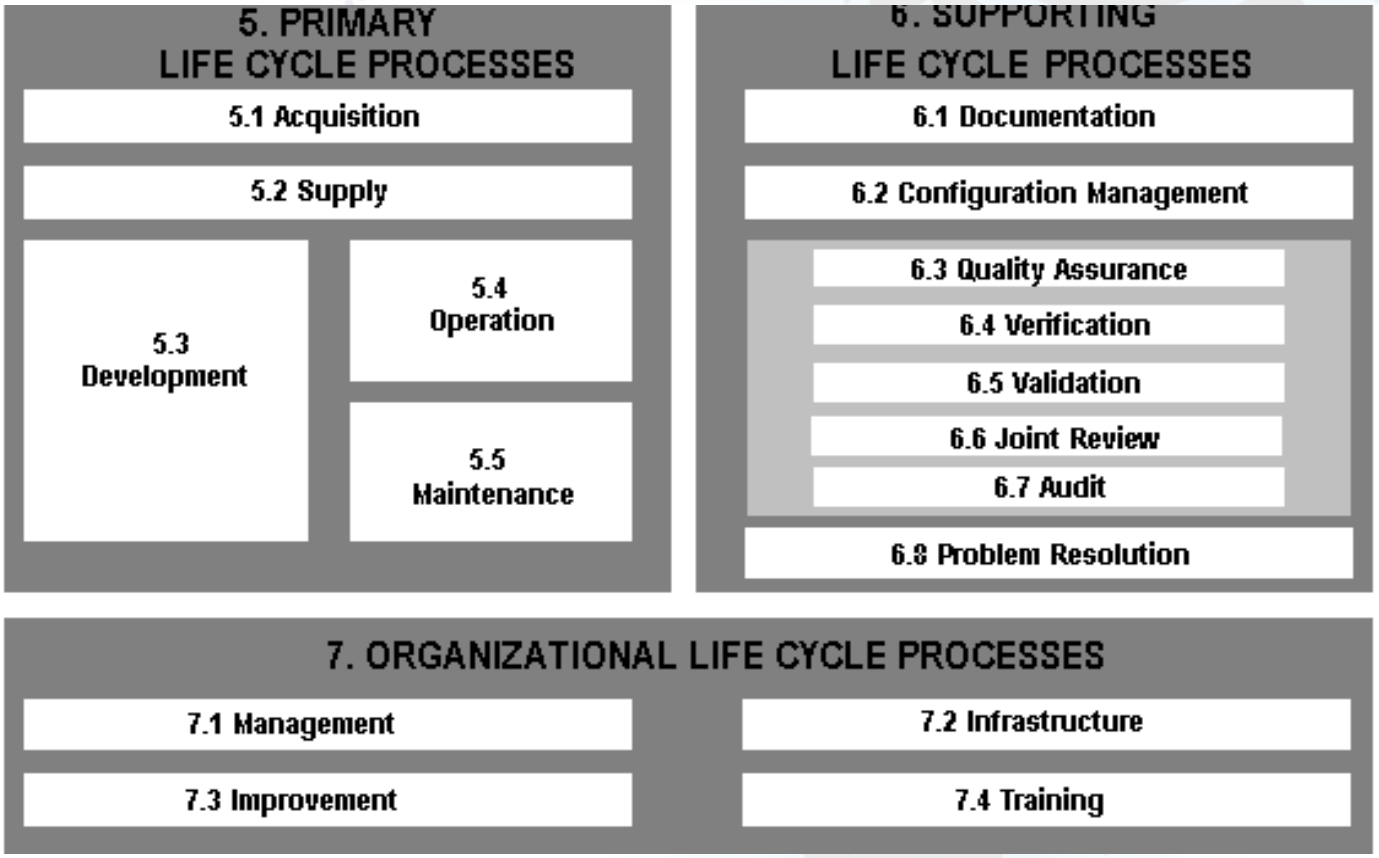
Common framework for software life cycle process

+

Well-defined terminology

- http://www.iso.org/iso/catalogue_detail?csnumber=43447

SE Overview



Overall structure of ISO/IEC 12207-2008

SE Overview

- SWEBoK (Software Engineering Body of Knowledge)
  - ISO/IEC TR 19759:2013
  - established SE as a recognized engineering discipline
  - promotes a consistent view of SE worldwide
  - provides a foundation for curriculum development and individual certification and licensing material

- https://www.computer.org/web/swebok/v3

Software Development Processes

- Goal
  - Splitting work into distinct phases/stages, organizing SE activities/tasks and deliverable/artifacts
- Software Development Process Models
  - Waterfall Model (1970s)
  - Spiral Model (1980s)
  - V model (1990s)
  - RUP (1990s)
  - Bazar Model (2000s)
  - Agile models (Scrum, XP, FDD) (2000s)

SE “Approaches”

- Software-product line
  - Multi software-product line
- Component-based SE
- Service-oriented SE
- Model-driven SE
- Agent-oriented SE
- OO SE and AO SE
- Theory-oriented SE
- Feature-oriented SE
- Others

SE Activities

Requirements

Design

- Software architecture

Construction/Implementation

Testing

Maintenance

Configuration management

Documentation

Validation and verification

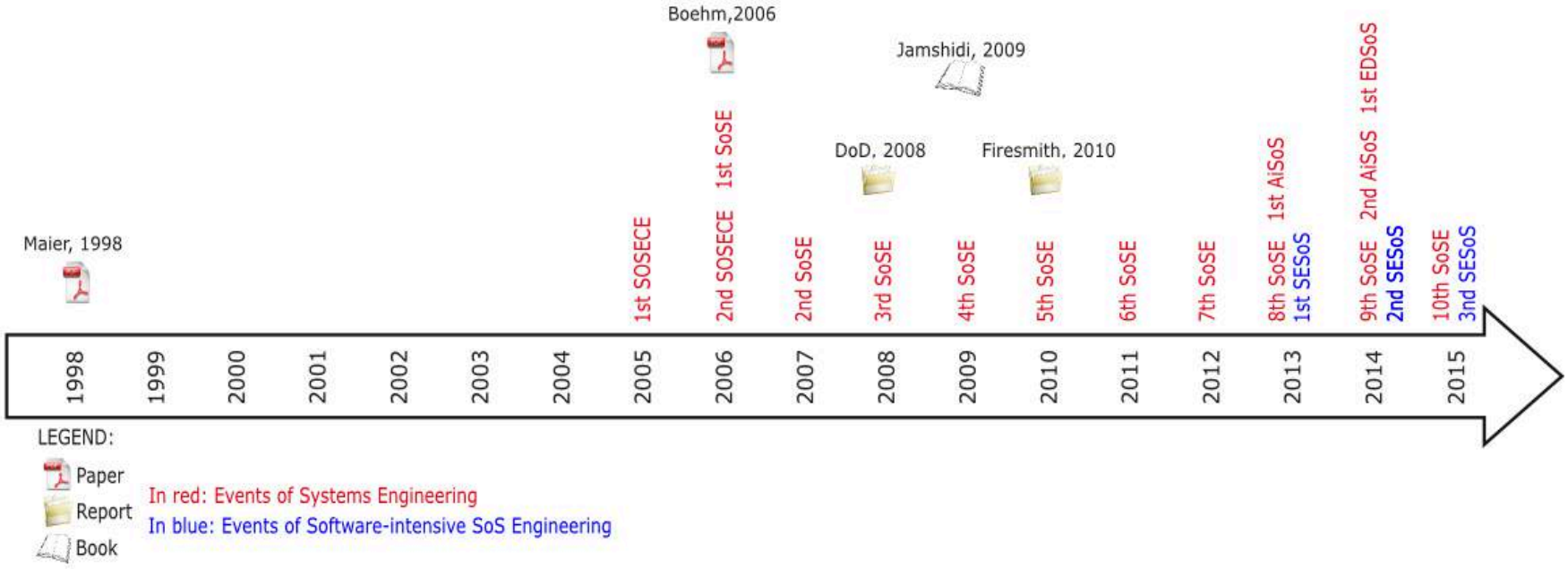
…

Standards, Patterns, Processes, Methods, Techniques, Tools

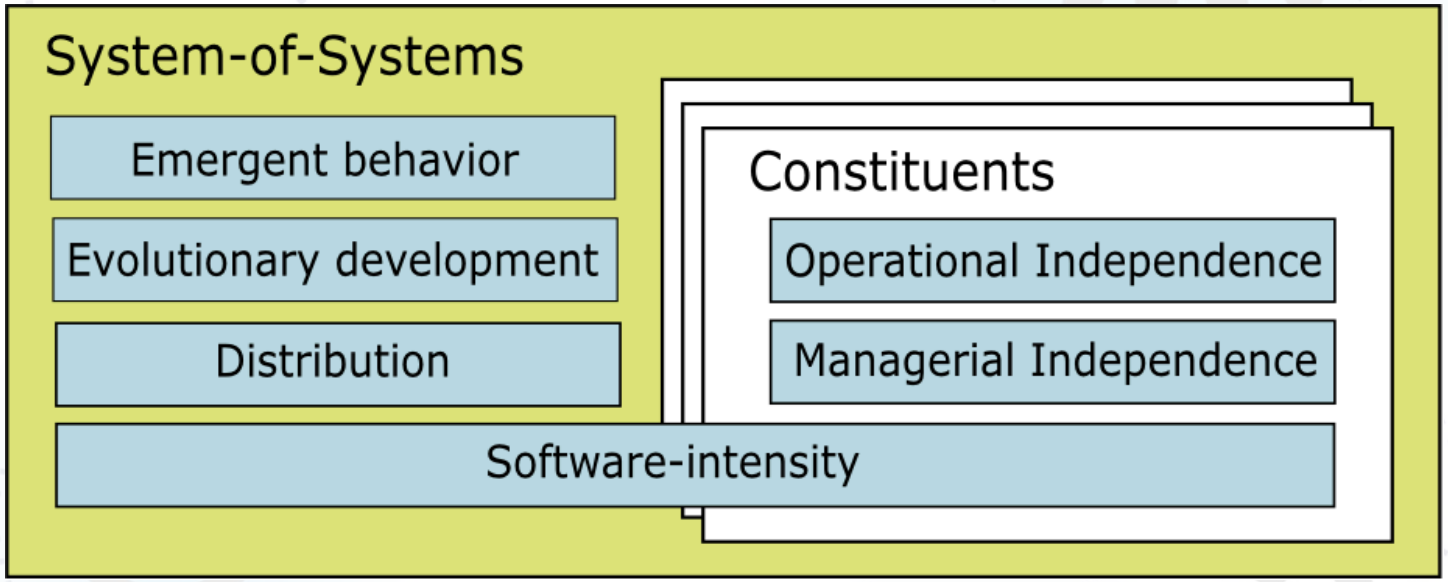
Experimental SE

Systems-of-Systems

Main Publications and Events



SoS Characteristics



SoS Characteristics

- **Characteristics related to the nature of constituents:**
  - *Operational independence*
    - Constituents operates independently, having its own mission and resources
  - *Managerial independence*
    - Constituents present independent management and evolve in ways not foreseen when they originally joined to particular SoS.

SoS Characteristics

- **Characteristics considering relationships among constituents:**
  - *Emergent behavior*
    - New behaviors from constituents
    - Behaviors non-predictable in design time emerge only at runtime
  - *Evolutionary development*
    - Constituents continually evolve, implying evolution in SoSs
    - SoSs evolve due to changes in their environment
  - *Distribution*
    - Distributed constituents, geographically or not
  - Software-intensity
    - Influence to the design, construction, deployment, and evolution of SoSs and constituents.
- **Consequence of SoS characteristics:**
  - **Dynamic architectures**
    - Changes at runtime in the overall structure
    - Changes in the relationship among constituents

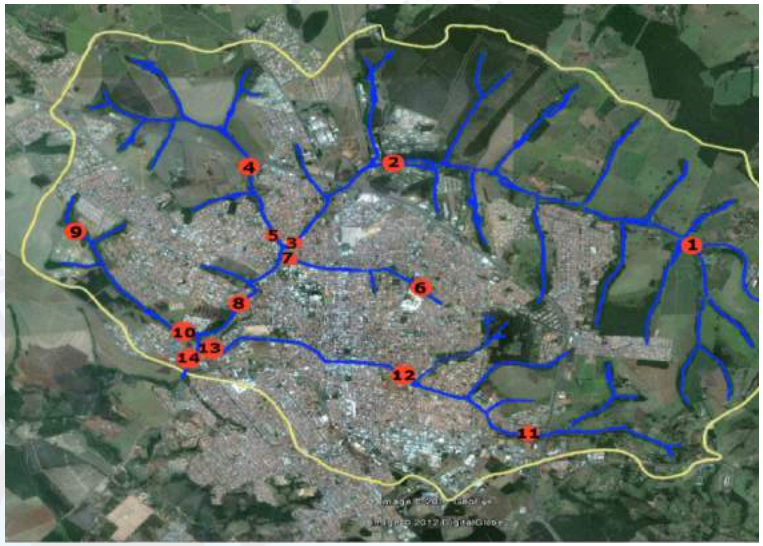
Examples

The New Frontier: Software-intensive Systems-of-Systems

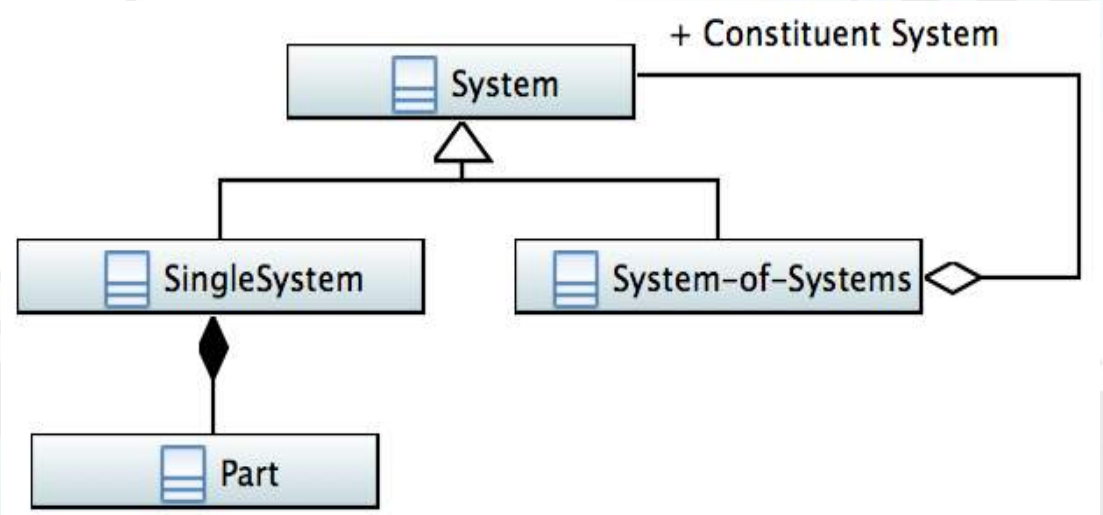
Smart-* Systems

Examples

- **Flood Monitoring System-of-Systems**
- Some possible constituent systems:
  - Wireless Sensor Networks (WSNs)
  - Unmanned Aerial Vehicles (UAVs)
  - Vehicular Ad-hoc Networks (VANETs)
  - Firefighter Systems
  - Weather Forecasting Systems
  - SMS Multicasting Systems
  - Mobile Apps in Smartphones
  - …



Relating Systems and SoS



- **A SoS is a system, too!!!**

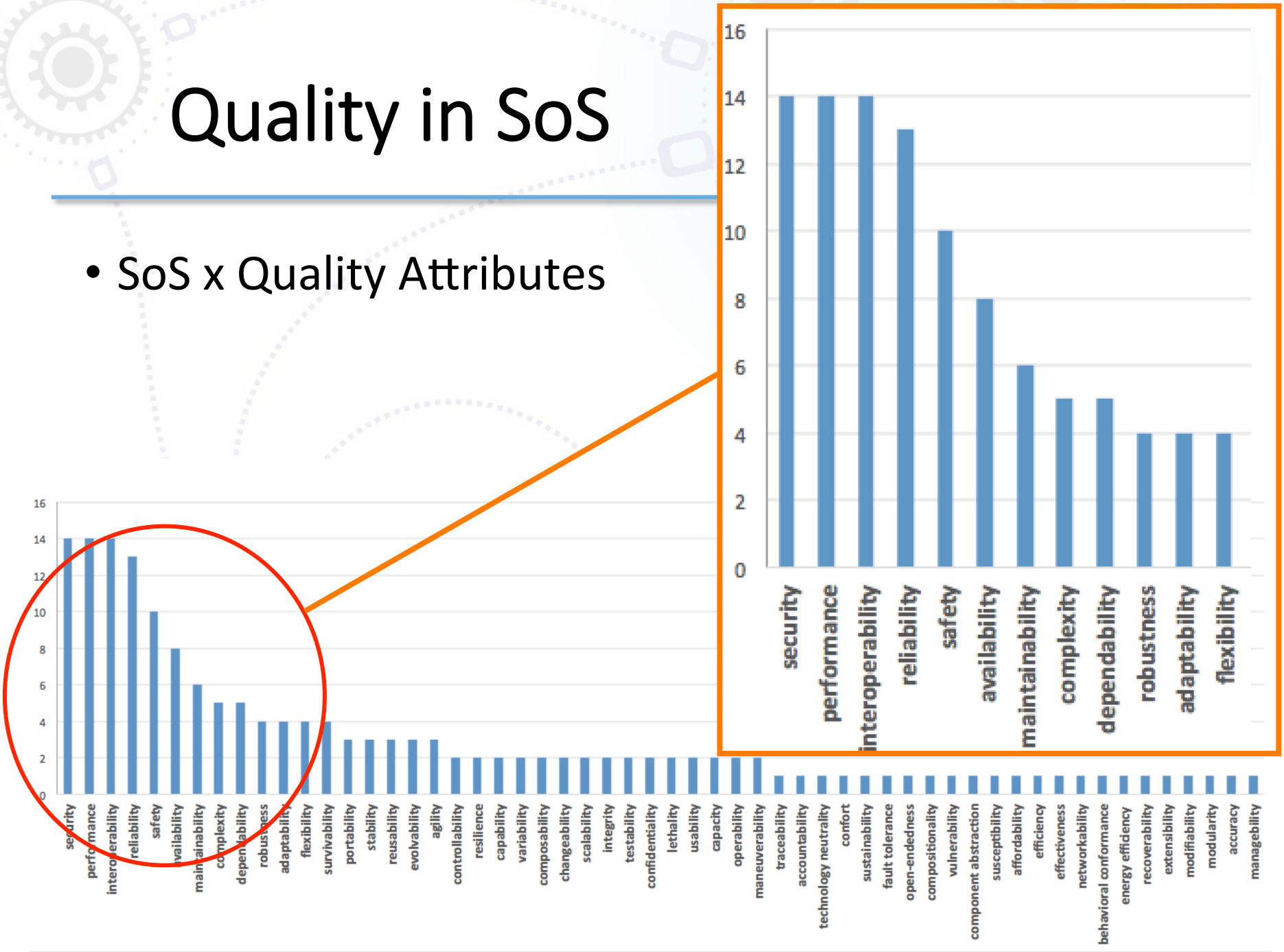
Definition

- **"System-of-Systems** is any system that:
  - results from the interoperation of organizational and managerial independent constituents, which have their individual mission and participate aware or not to comply with a global mission;
  - has evolutionary development resulting from evolution of constituents and/or changes in the environment;
  - presents emergent behaviors, expected or non-expected in design time, resulting from the interaction among constituents at runtime; and
  - depends on software as an enabling technology to its design and evolutionary development."

[Nakagawa, Oquendo, Maldonado, 2016]

Quality in SoS

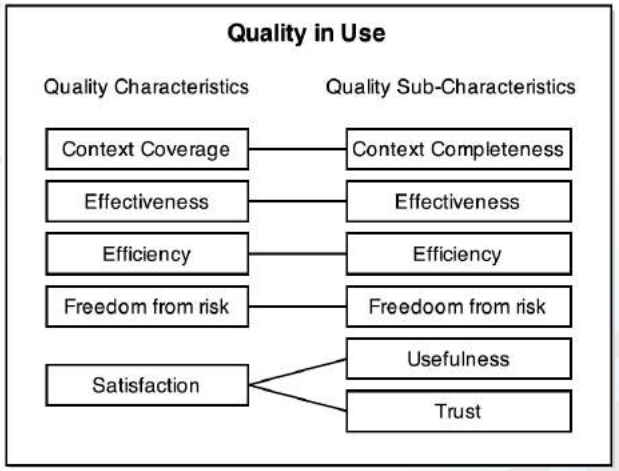
- SoS x Quality Attributes

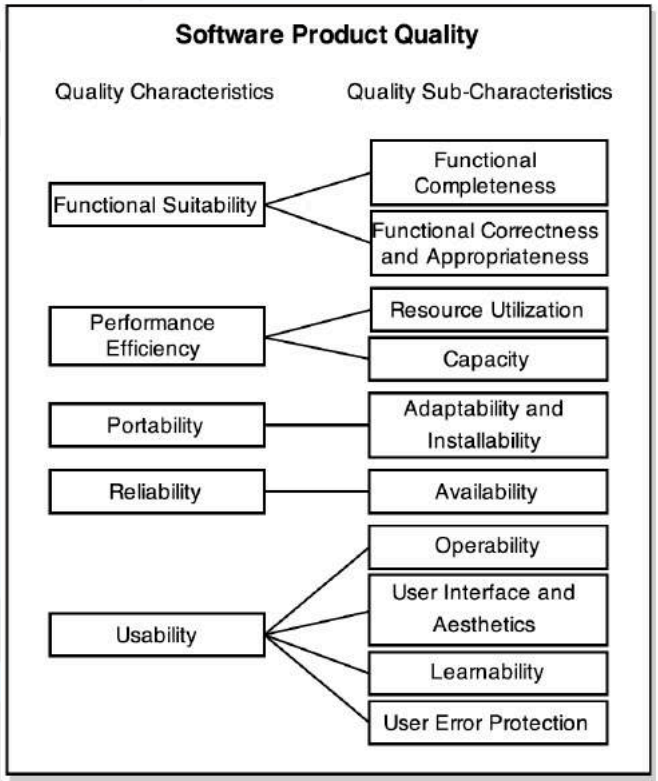


Quality in SoS

- **Essential factor in SoS**
  - **Most of SoS address critical domain**

- **Quality model for crisis/ emergent management domain**

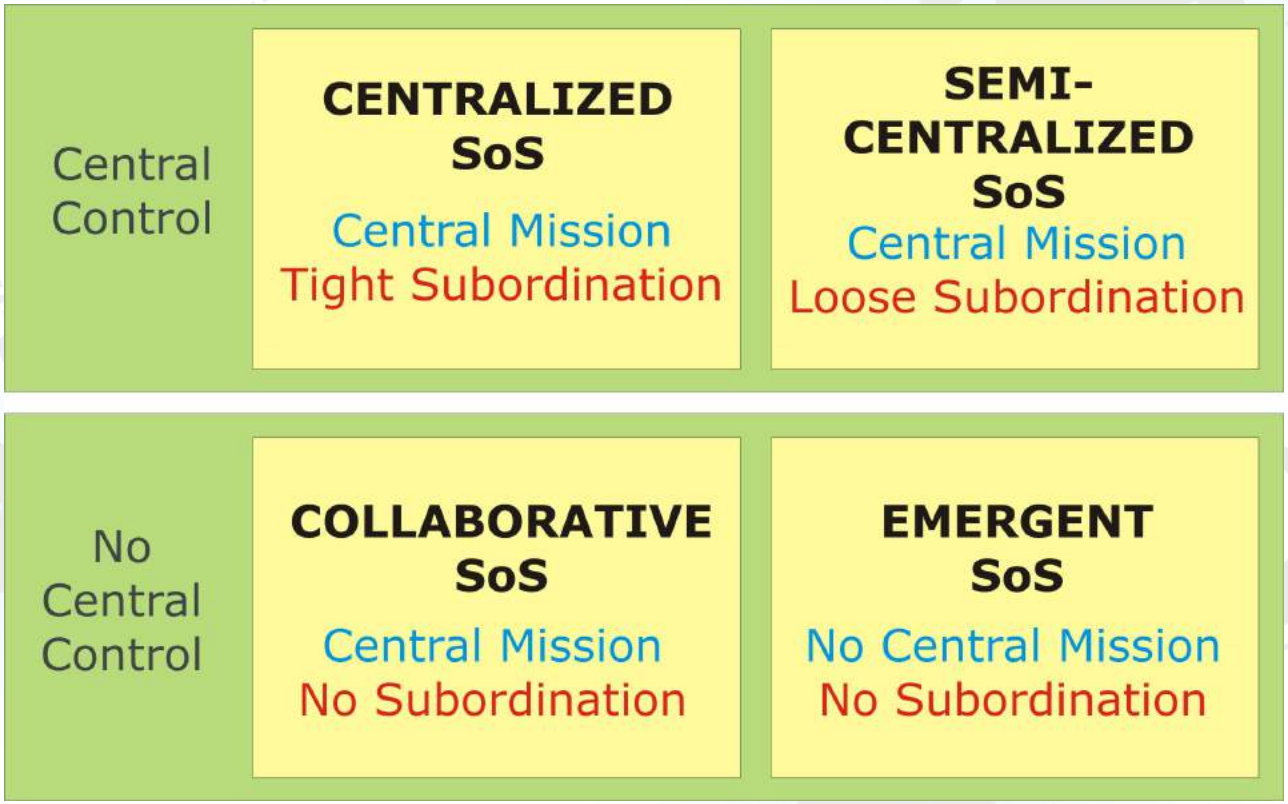




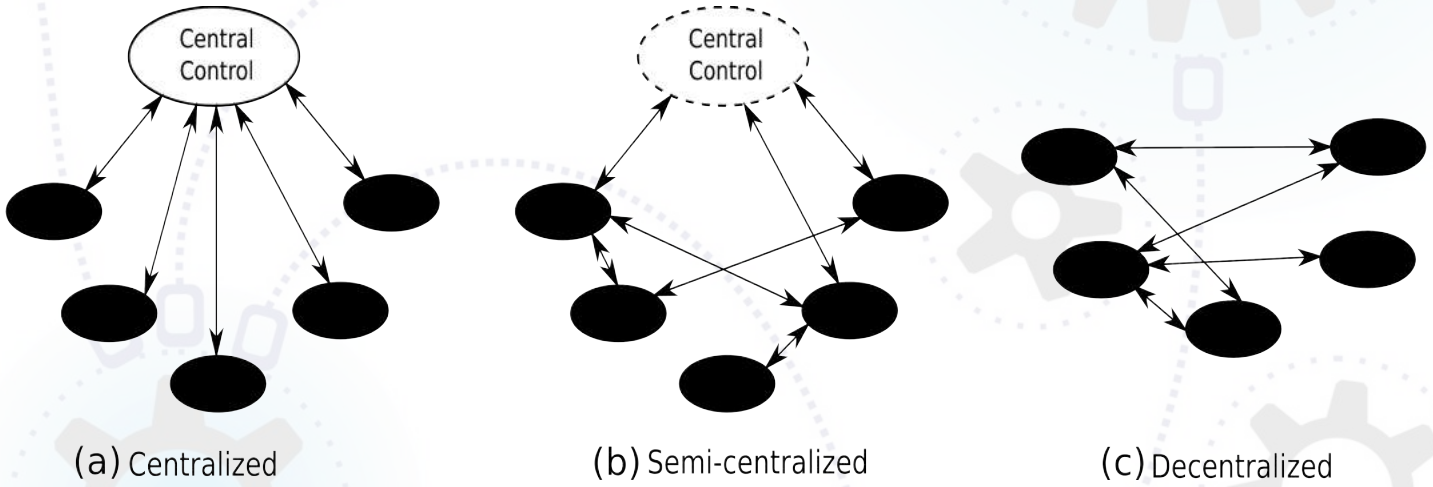
Types: Tentatives

- Directed SoS
  - SoS that are centrally managed
  - Constituents are especially developed or acquired to fit specific purpose
  - Constituents operate under tight subordination
- Acknowledged SoS
  - SoS that are centrally managed
  - Constituents retain their operational independence
  - Constituents operate under loose subordination
- Collaborative SoS
  - There is no central management
  - Constituent systems voluntarily agree to fulfill central purposes
- Virtual SoS
  - There is no central authority or centrally agreed purpose

Types: Tentatives



Types: Tentatives



(a) Centralized (b) Semi-centralized (c) Decentralized

Challenges/Questions about SoS Development and Evolution

- Do traditional SE work on SoS?
  - What works? What does not work? What needs adaptation?
- How to manage SoS evolution?
  - How to manage the SoS emergent behaviors?
  - How to manage the SoS dynamic architectures?

- One of the solutions: **Software Architecture**

SoS Software Architecture

- Software architectures
  - Backbone for software-intensive systems
  - Fundamental in determining the system quality
  - Considerable amount of research, mainly regarding their design, representation, and evaluation.

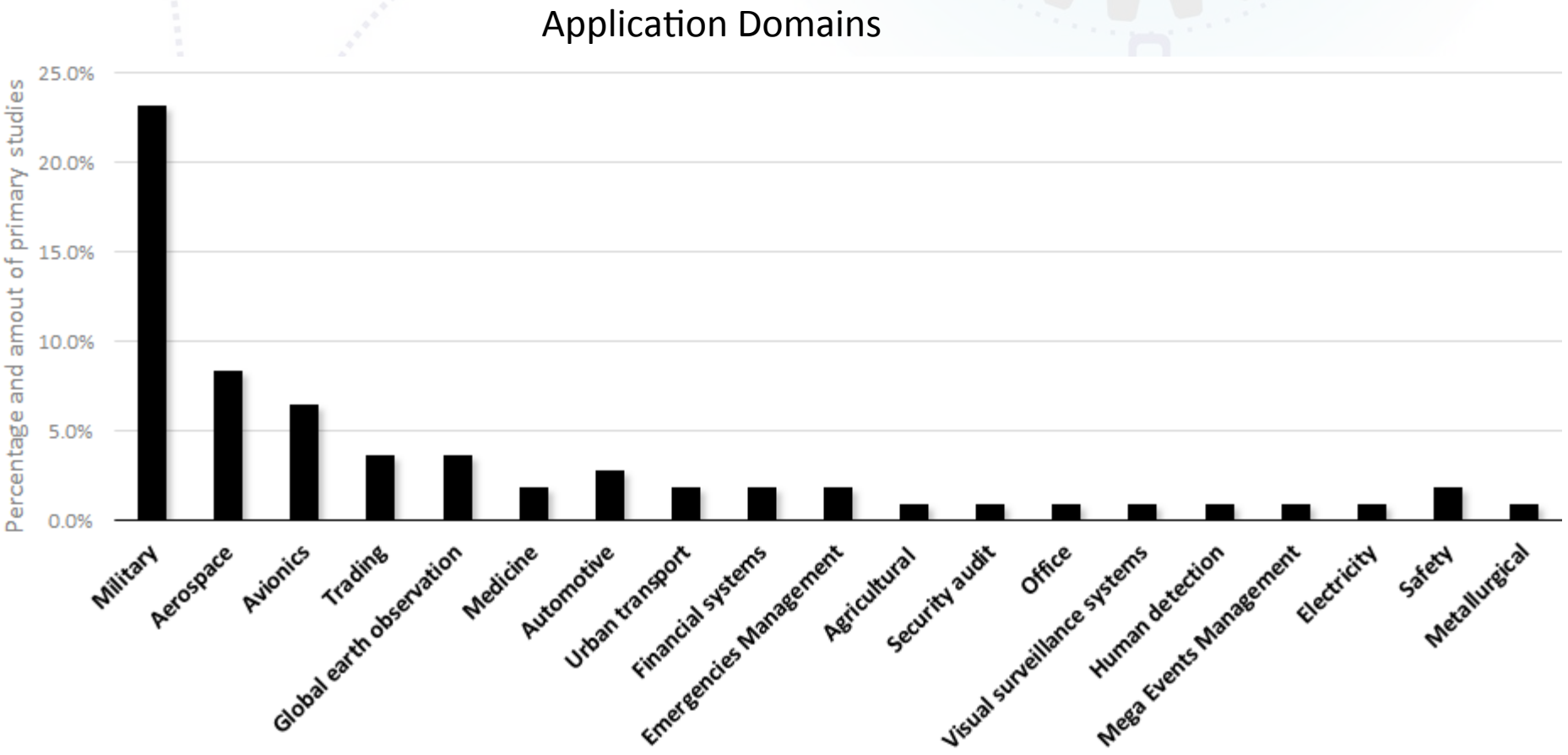
- Software architectures for SoS is a new, important research area!!

SoS Software Architecture

- "The software architecture of a SoS is a dynamic structure or structures of a system, which comprise the independent constituent systems, the externally visible properties of those constituents, the relationships among them, and the principles and constraints that guide both its initial design and its evolution imposed by the emergence of expected and non-expected missions at runtime."

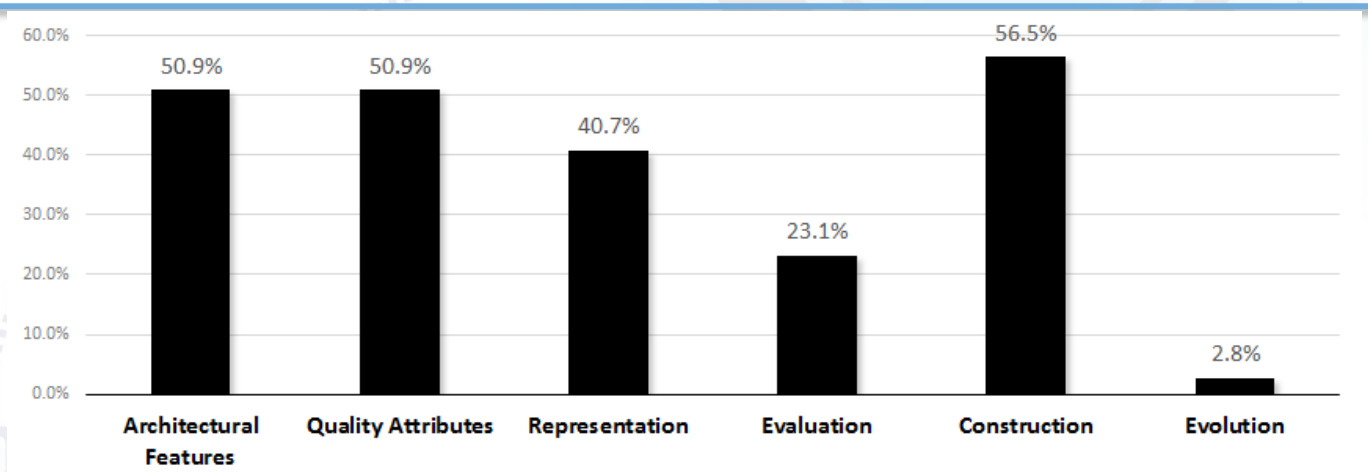
[Nakagawa, Oquendo, Maldonado, 2016]

SoS Software Architecture

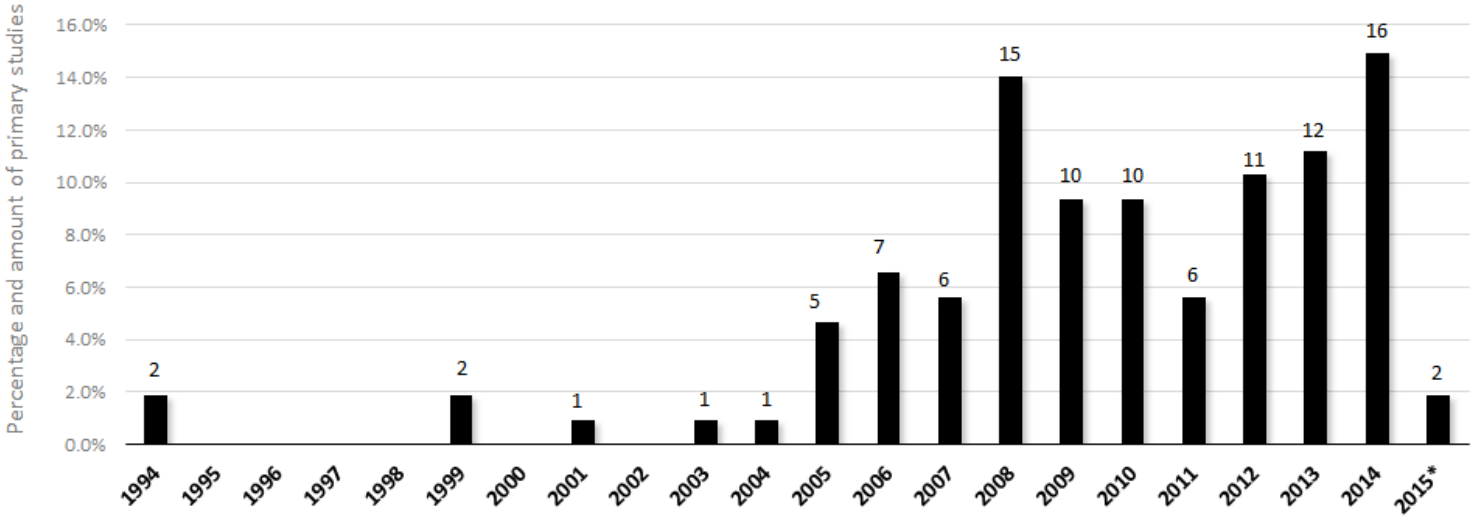


SoS Software Architecture

Research Topics on SoS Software Architecture



Publication Distribution through Years



*only studies found until july 2015

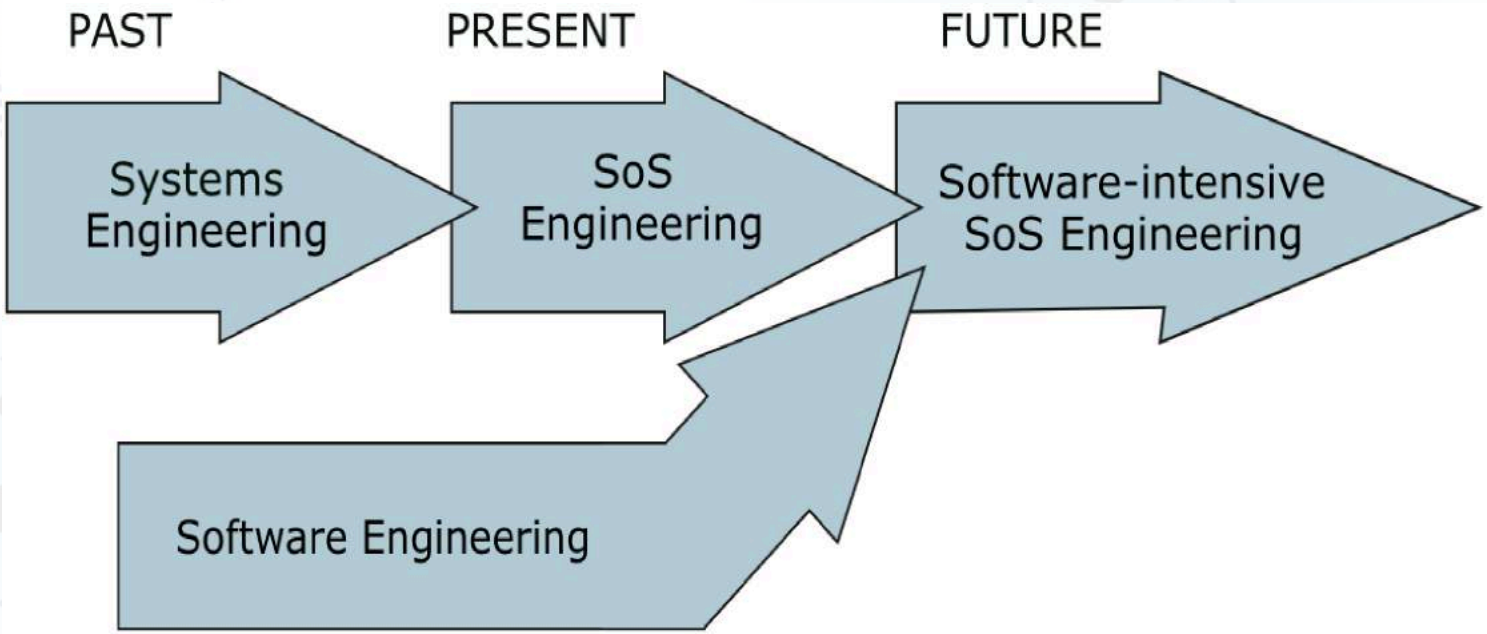
Main Materials on SoS

- B. Boehm and L.A. Lane, “21st Century Processes for Acquiring 21st Century Software-Intensive Systems of Systems,” The Journal of Defense Software Engineering, vol. 19, no. 5, 2006, pp. 4-9.
- Department of Defense, “Systems Engineering Guide for Systems of Systems,” Aug. 2008; www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf.
- D. Firesmith, “Profiling Systems Using the Defining Characteristics of Systems of Systems (SoS)”, tech. report CMU/SEI-2010-TN-001, Software Engineering Institute, Carnegie Mellon Univ., 2010.
- M. Jamshidi, ed., “System of Systems Engineering: Innovations for the Twenty-first Century”, Wiley & Sons, 2009, p. 616.
- M.W. Maier, “Architecting Principles for Systems-of-Systems,” Systems Engineering, vol. 1, no. 4, 1998, pp. 267-284.
- C. B. Nielsen, P. G. Larsen, J. Fitzgerald, J. Woodcock, and J. Peleska, “Systems of systems engineering: Basic concepts, model-based techniques, and research directions,” ACM Computing Survey, 48, 2, 2015.

Conclusion and Future Work

- System Engineering
  - Great contribution to SE and SoSSE
- SE
  - More into the software industry
  - Evolution to software-intensive SoS
- Several lines of research in SoS
  - Software architecture and reference architecture
  - Software testing
  - Multi-product line
  - Sustainability/longevity
    - http://www.journals.elsevier.com/journal-of-systems-and-software/call-for-papers/special-issue-on-sustainability-and-longevity-of-systems
  - Several others

Conclusions and Future Work



From Systems Engineering to Software Engineering: A Perspective for Systems-of-Systems

Prof. Dr. Elisa Yumi Nakagawa

Prof. Dr. José Carlos Maldonado

April, 2016