Introduction to System Modeling

Indriani Noor H.

Prepared for CSP320 System Modeling
Faculty of Computer Science
Universitas Esa Unggul - 2018

***References**: International Council on Systems Engineering (INCOSE), 2006*

Why System Modeling?

- Modeling, simulation, and prototyping used during architecture design **can significantly reduce the risk of failure in the finished system**.
- Modeling **helps generate data** in the domain of the analyst or reviewer, not available from existing sources, in a manner **that is affordable and timely to support**.
- These techniques enable the development of complex and costly enabling systems, such as a flight simulator or a high-volume production line, which allow validation of the system's concepts, or supports training of personnel in ways that would otherwise **be cost prohibitive**.
- Systems engineers use modeling and simulation on large complex projects to **manage the risk of failure to meet system mission and performance requirements**.
- This form of analysis is best conducted by subject matter experts who **develop** and **validate** the models, conduct the **simulations**, and **analyze** the **results**.

(International Council on Systems Engineering, 2006)

Interdisciplinary perspective on how to design and manage complex systems over their life cycles

- System Engineeri ng

- Software Engineeri ng

Closing the gap between systems and software engineering

What is System?

- Systems *are man-made, created and utilized to provide services in defined environments for the benefit of users and other stakeholders.*
- *These systems may be configured with one or more of the following: hardware, software, humans, processes (e.g., review process), procedures (e.g., operator instructions), facilities, and naturally occurring entities (e.g., water, organisms, minerals).*
- *In practice, they are thought of as <u>products</u> or <u>services</u>.*

(ISO/IEC 15288)

System

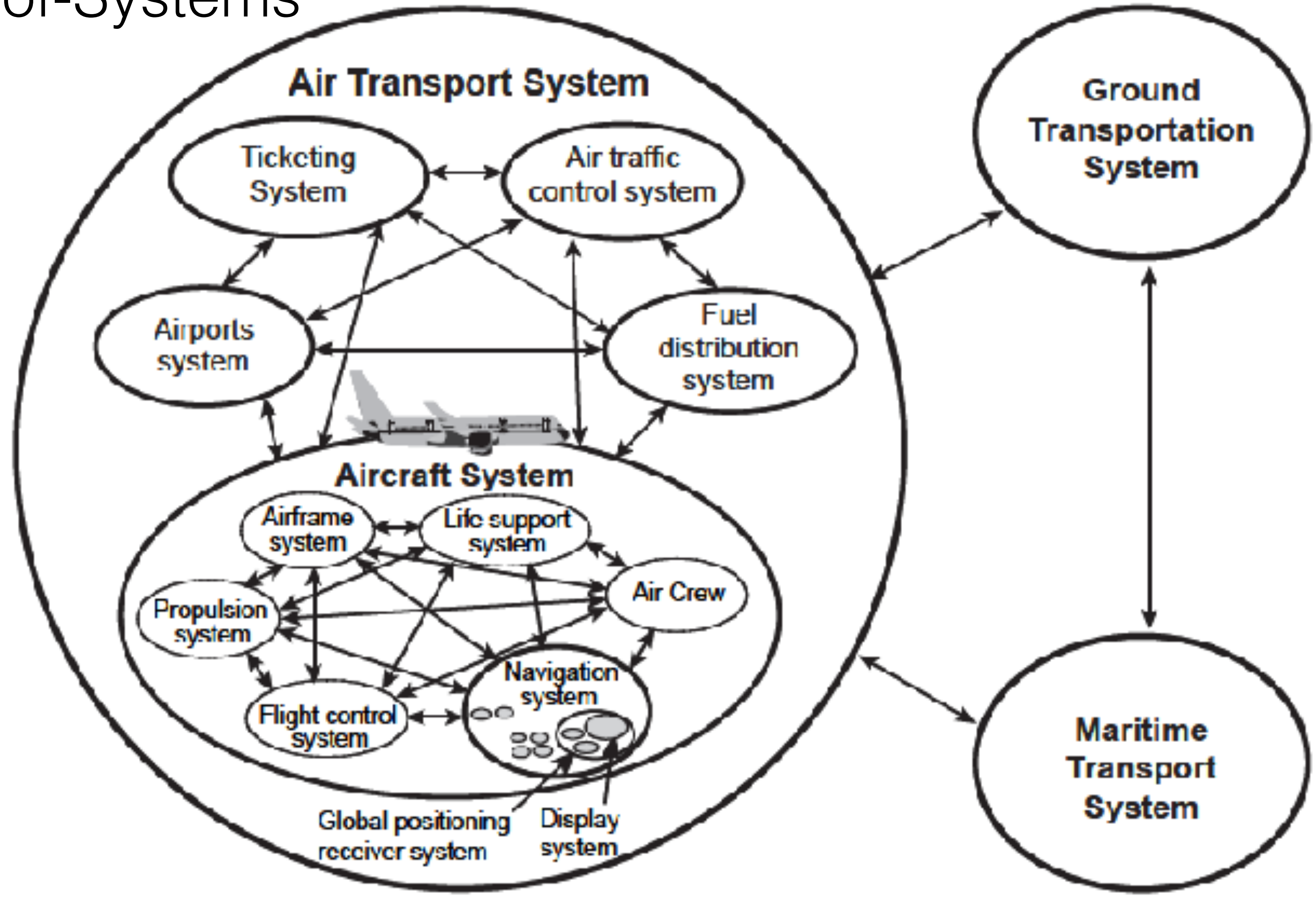
- **A system boundary** defines the scope of a system, creating a distinction between the system and the environment, or context, in which a system exists.
- **A system architecture** is defined as, quote, the fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution, unquote. A system is made up of individual elements.
- System elements can be
  - **atomic**, meaning they cannot be broken down further, or can be systems in their own right.
  - **decomposable** into further subsystem elements.

Systems of Systems (SoS)

- "Systems-of-Systems" (SoS) are defined as an interoperating collection of component systems that produce results unachievable by the individual systems alone
  (International Council on Systems Engineering, 2006)
- *The perception and definition of a particular system, its architecture and its system elements depend on an observer's interests and responsibilities.*
  (ISO/IEC 15288)
  - *One person's system-of-interest can be viewed as a system element in another person's system-of-interest.*
  - *Conversely, it can be viewed as being part of the environment of operation for another person's system-of-interest.*

International Council on Systems Engineering (2006)

Example of the multitude of perceivable systems-of-interest in an aircraft and its environment of operation within a Transport System-of-Systems



Challenges in the Development of SoS

1. **System elements operate independently.** Each system in a system of systems is likely to be operational in its own right.
2. **System elements have different life cycles.** SoS involves more than one system element. Some of the system elements are possibly in their development life cycle while others are already deployed as operational. In extreme cases, older systems elements in SoS might be scheduled for disposal before newer system elements are deployed.
3. **The initial requirements are likely to be ambiguous.** The requirements for a system of systems can be very explicit for deployed system elements. But for system elements that are still in the design stage, the requirements are usually no more explicit than the system element requirements. Requirements for SoS mature as the system elements mature.

Challenges in the Development of SoS (Cont.d)

4. **Complexity is a major issue.** As system elements are added, the complexity of system interaction grows in a non-linear fashion. Furthermore, conflicting or missing interface standards can make it hard to define data exchanges across system element interfaces.
5. **Management can overshadow engineering.** Since each system element has its own product/project office, the coordination of requirements, budget constraints, schedules, interfaces, and technology upgrades further complicate the development of SoS.
6. **Fuzzy boundaries cause confusion.** Unless someone defines and controls the scope of a SoS and manages the boundaries of system elements, no one controls the definition of the external interfaces.
7. **SoS engineering is never finished.** Even after all system elements of a SoS are deployed, product/project management must continue to account for changes in the various system element life cycles, such as new technologies that impact one or more system elements, and normal system replacement due to preplanned product improvement.