Managing Product Information for Complex Systems

Adopting MBSE in the Joint Strike Missile (JSM) project

KONGSBERG

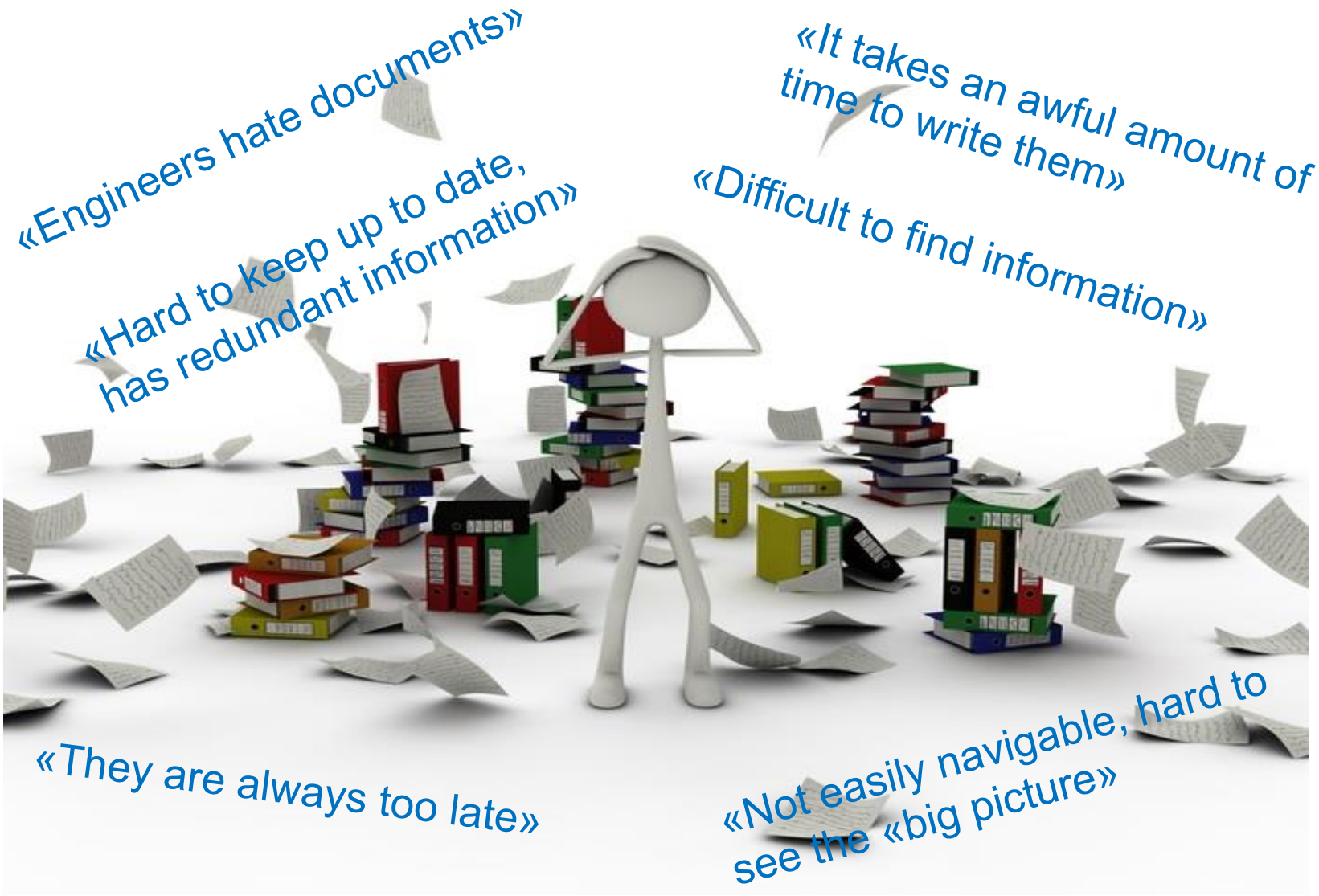
Svein Erik Søgård
Kongsberg Defence Systems

About Presenter

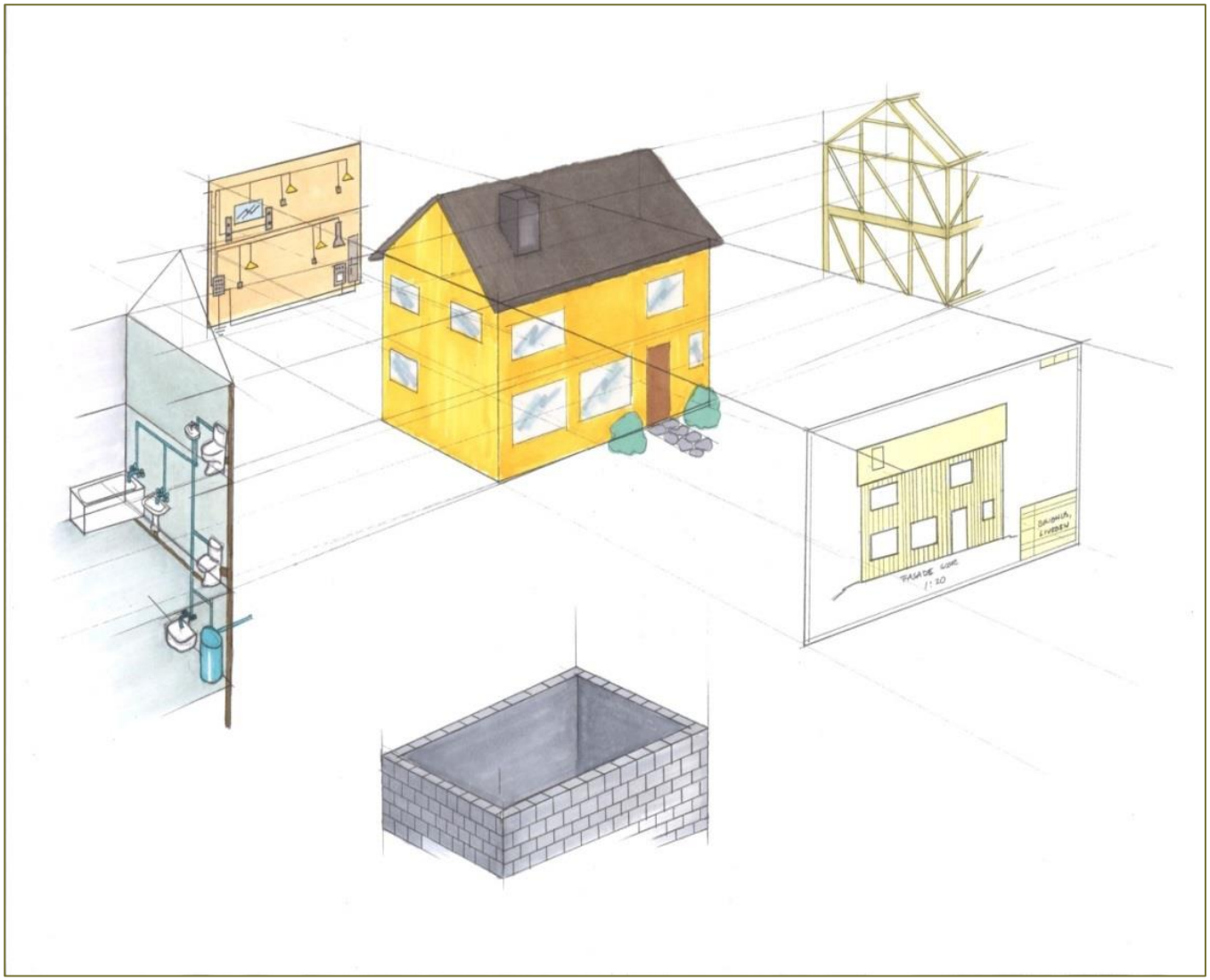
- Svein Erik Søgård, MSc
  - Principal Engineer, Missile Systems

- Engaged since 1995 at Missile Division in KDS
- Background from SW development, system integration and test in NSM (Naval Strike Missile)
- Current work (since 2010) : system architecture in JSM and responsible for adopting MBSE using SysML

Motivation: The problem with document based communication

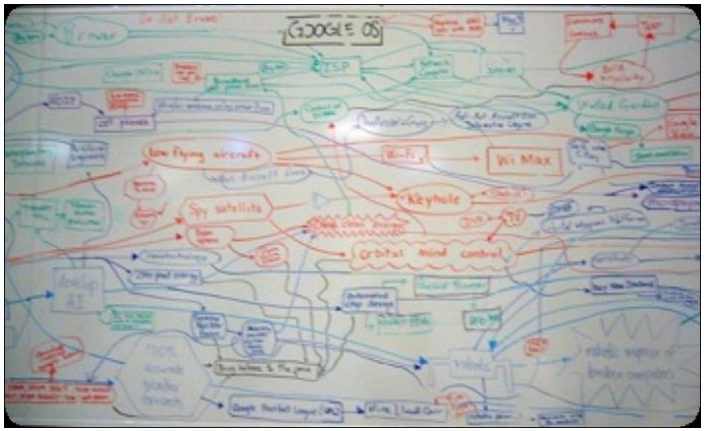


Solution: Model Based approach with views



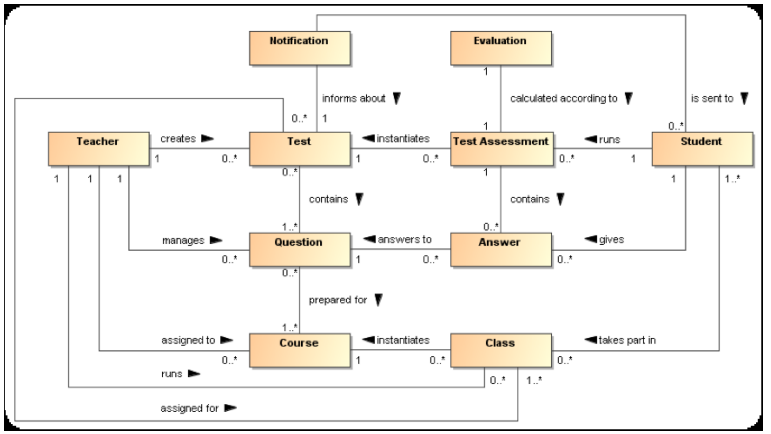
What is a model? (in this context)

Drawing



≠

Model



- Model
  - Integrated information model built up of model elements that are related to each other
  - Strict rules for what kind of elements and which relations that can be applied. Rules are validated by a tool
- Diagrams are not standalone drawings, they are visualizing the elements and the relations the model
- Single source of information: Update on an element is automatically reflected in all views/diagrams

The MBSE approach in JSM

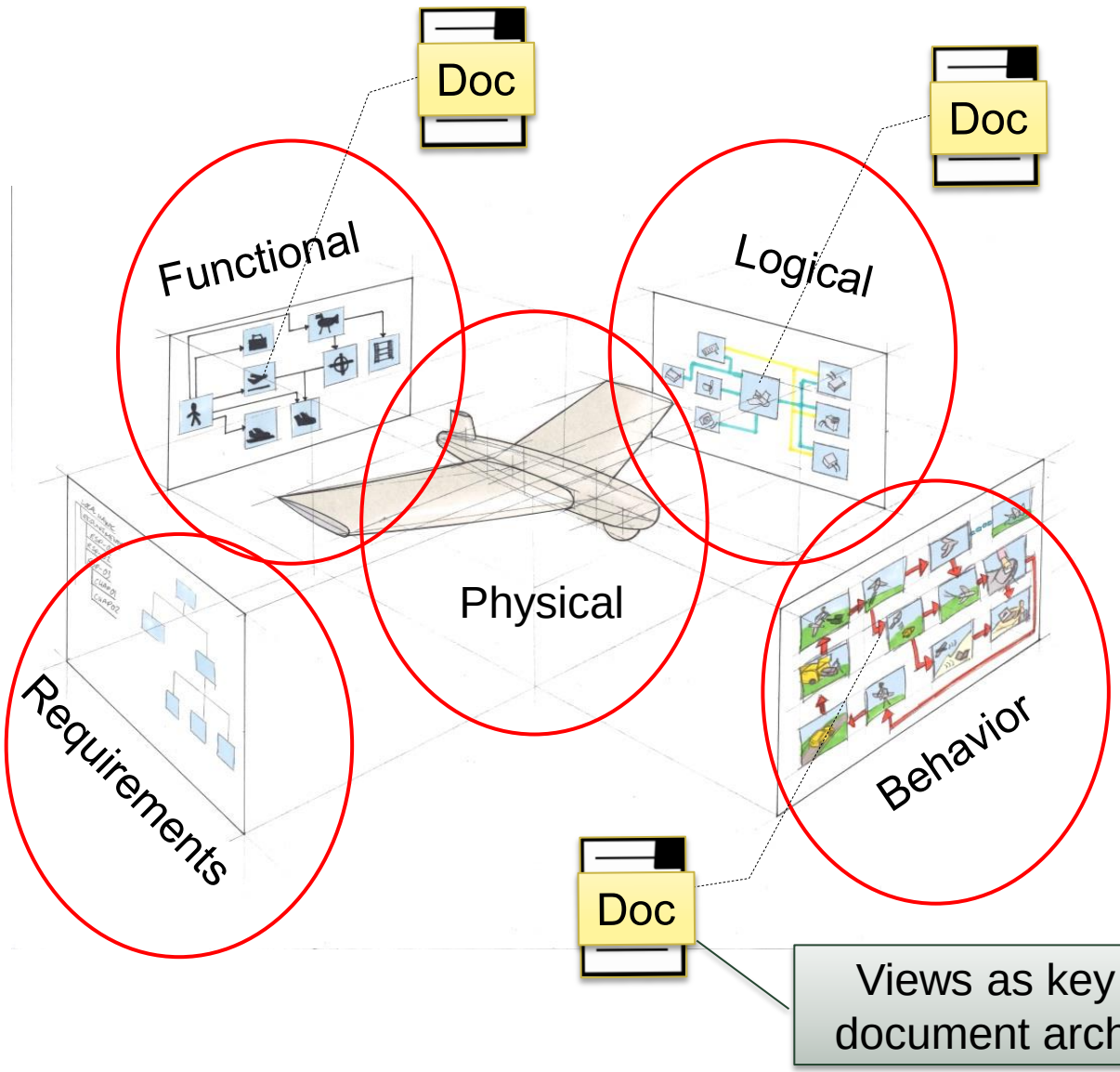
KONGSBERG

JSM – some key characteristics

- Many different technologies to be integrated (multi disciplinary):
  - Infrared imaging target seeker
  - multi-sensor navigation system,
  - jet engine and bank-to-turn flight control
  - In-flight radio communication (weapon data link)
  - on-board mission/flight route planning based on operational scenario
  - programmable fuze/warhead
  - multicore computing platform
  - ……
- SW intensive
  - >60% of the system requirements affects SW
- >30 years product lifecycle, mid life updates

How do we manage this?

System Architecture Framework - Views



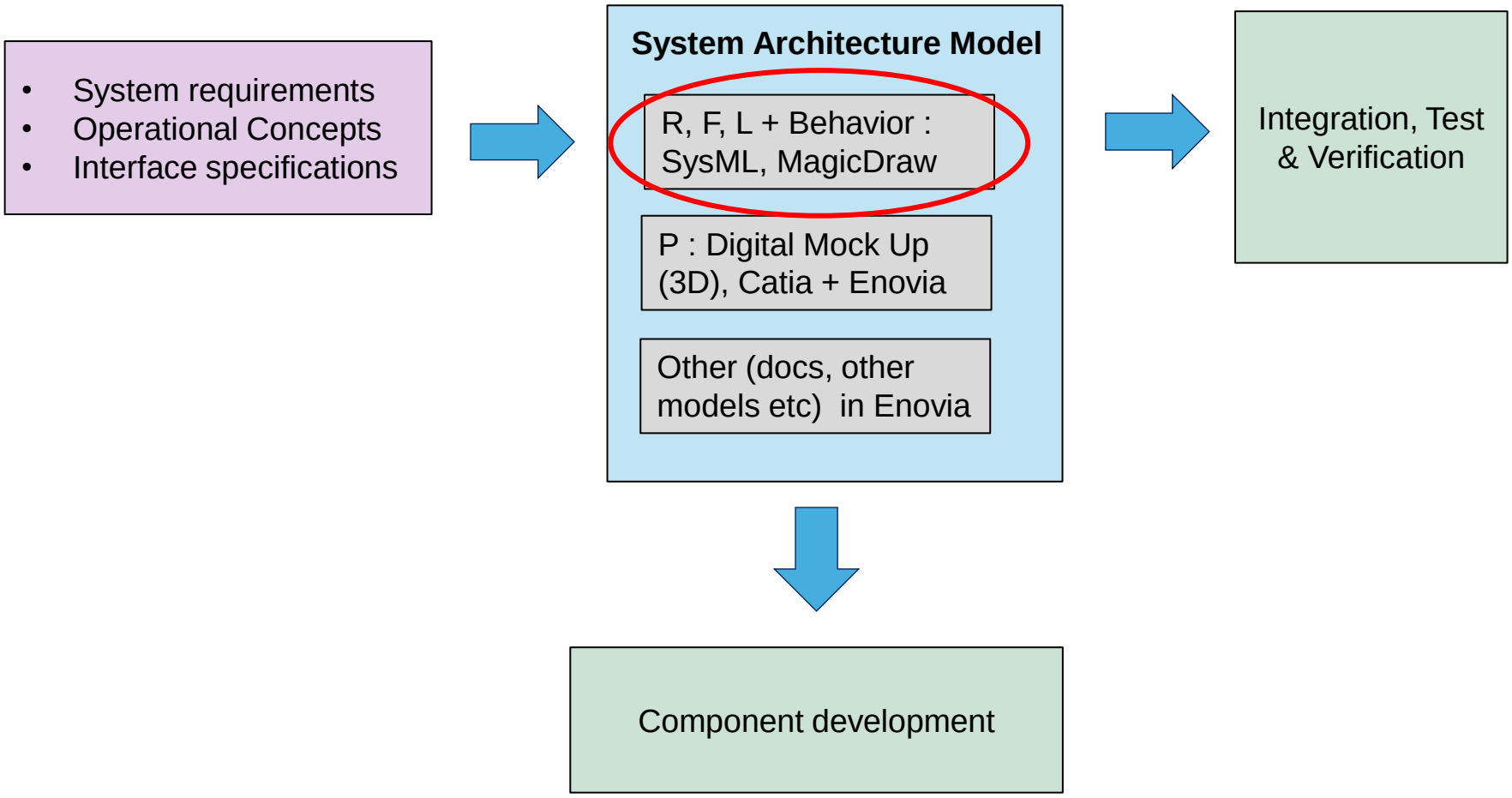
Specification
- WHAT shall the system do?
- Requirements and Behavior
- Context/interfaces

Functional Design:
- Given concept, HOW shall the system work?

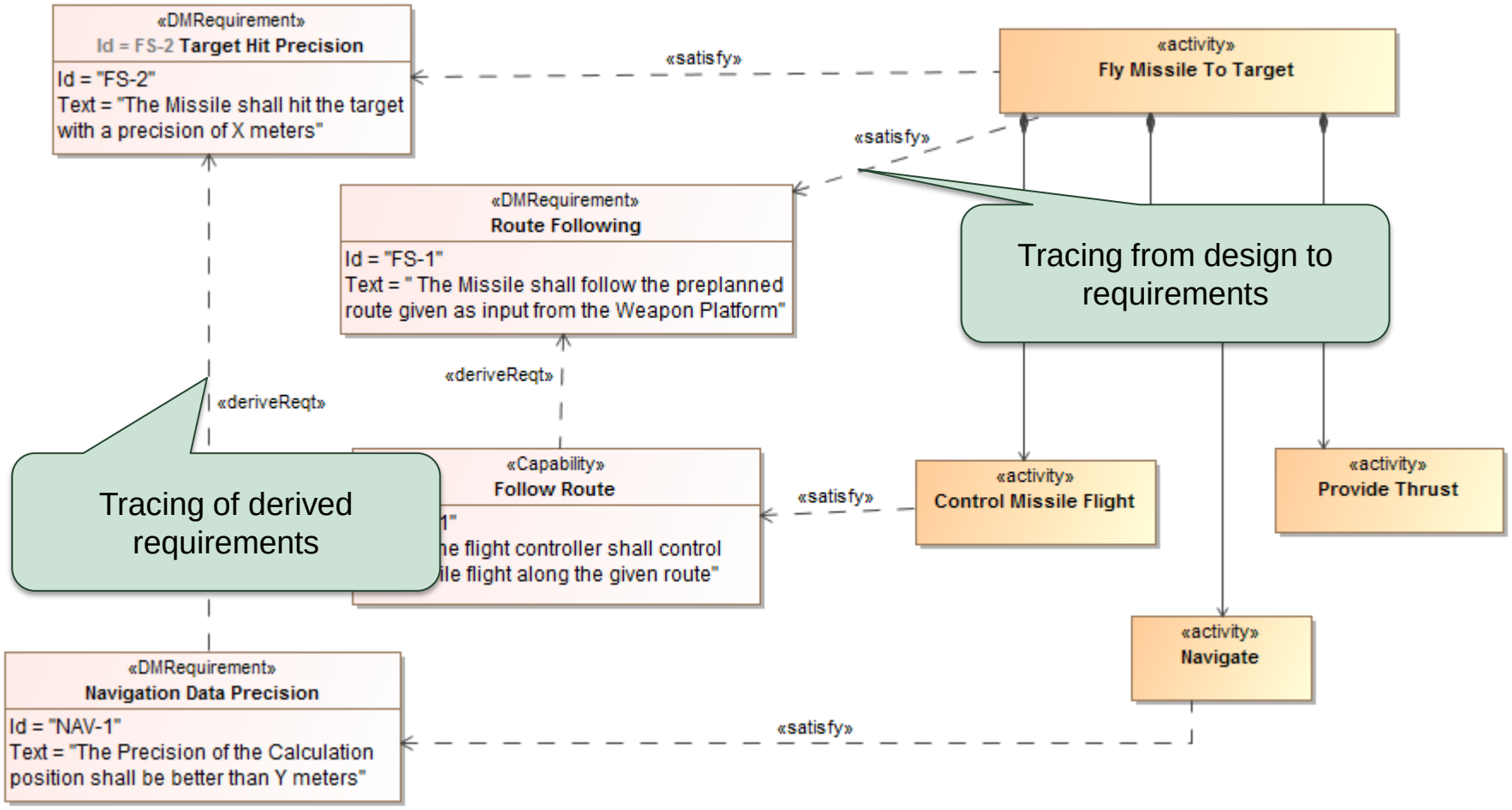
Logical Design
- Realization oriented
- HOW are components connected?
- HOW do they interact? (sequences/flow)?

Physical Design:
- Mechanical, 3D
- HOW is the product assembled?

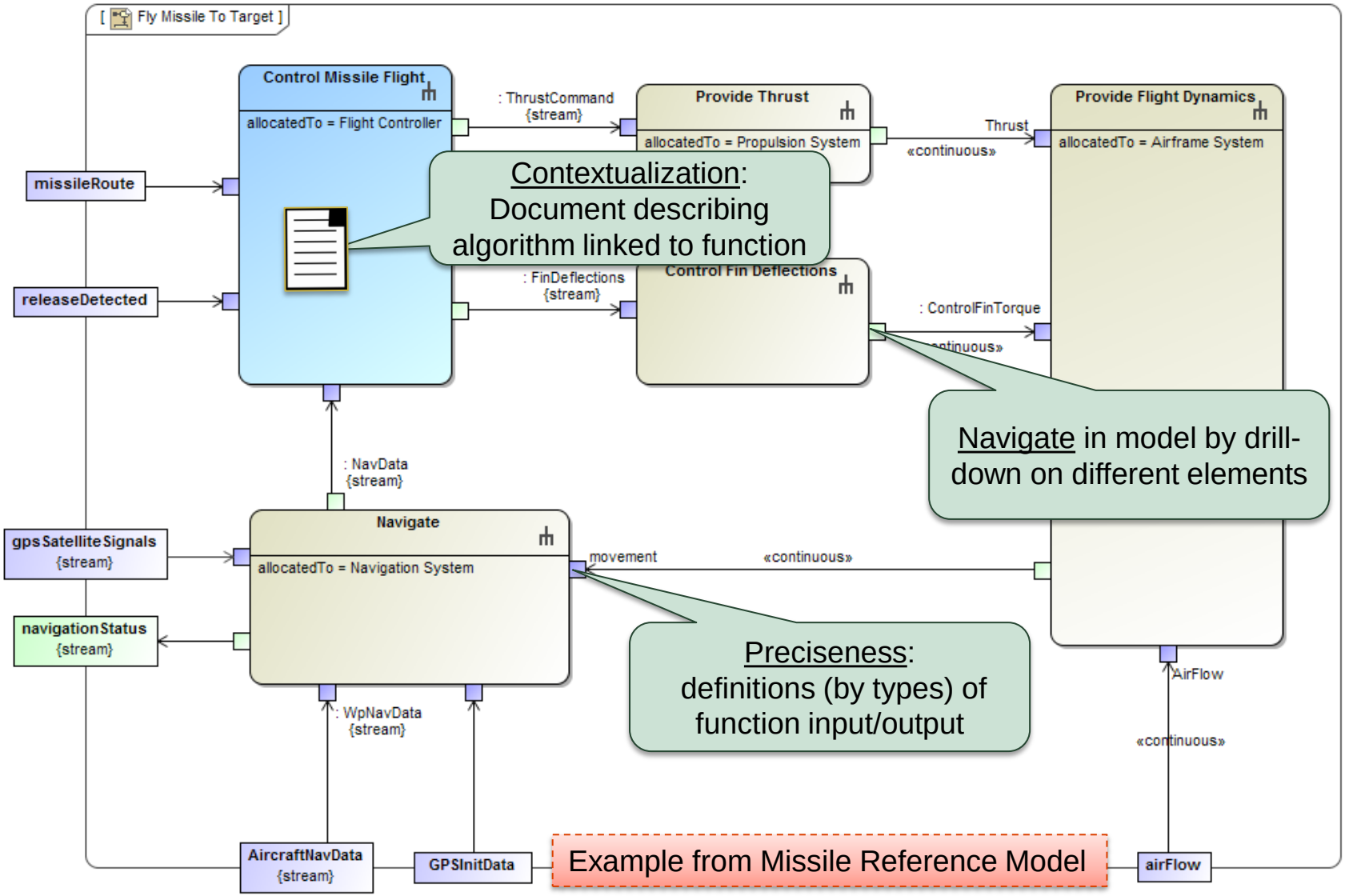
JSM System Architecture Model – Context



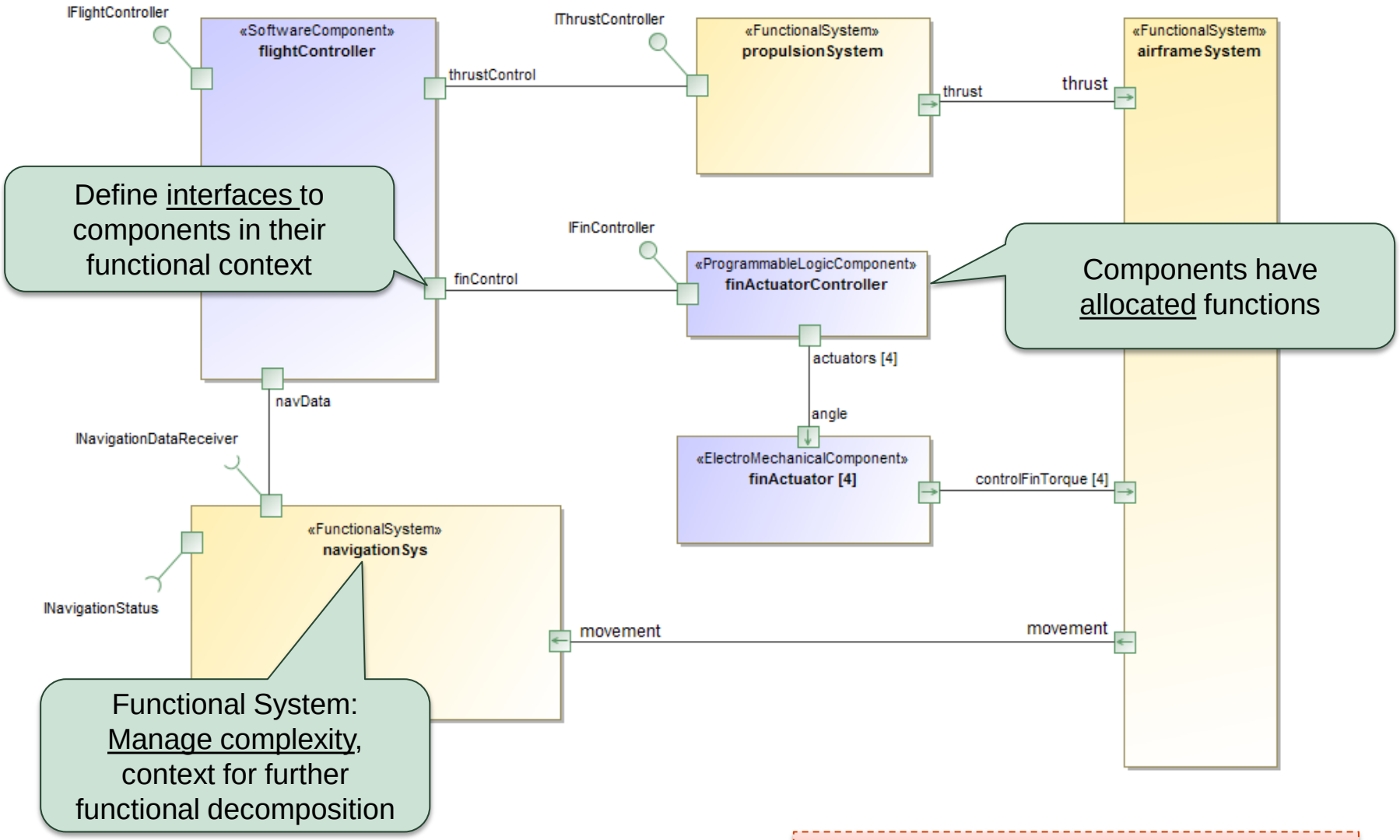
Functional Design - Requirements



Example – Flight System – Functional view



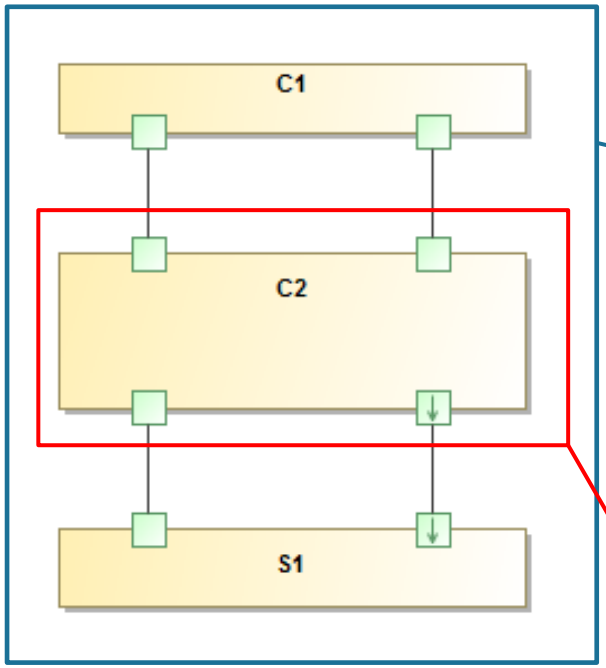
Example – Flight System – Logical design view

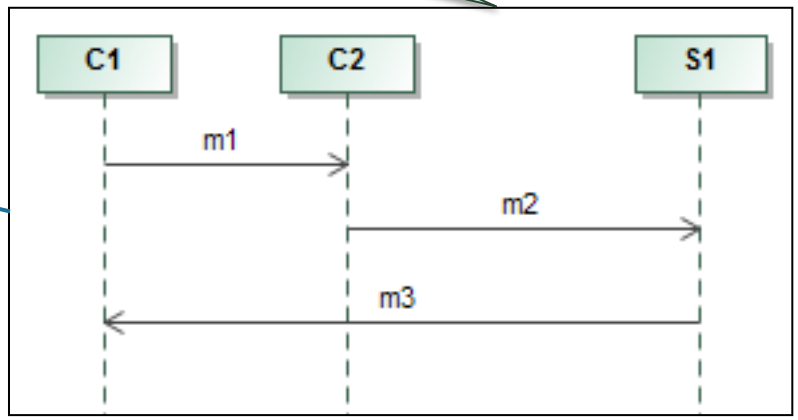


Example from Missile Reference Model.

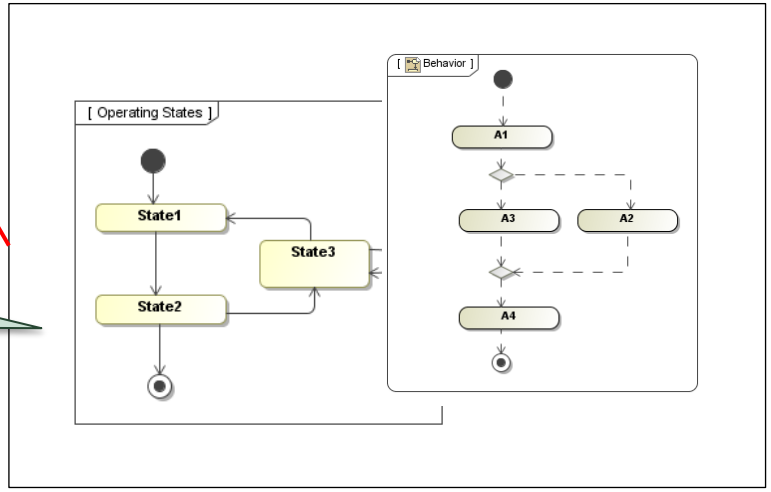
Defining Behavior

Sequence diagrams shows interactions between components

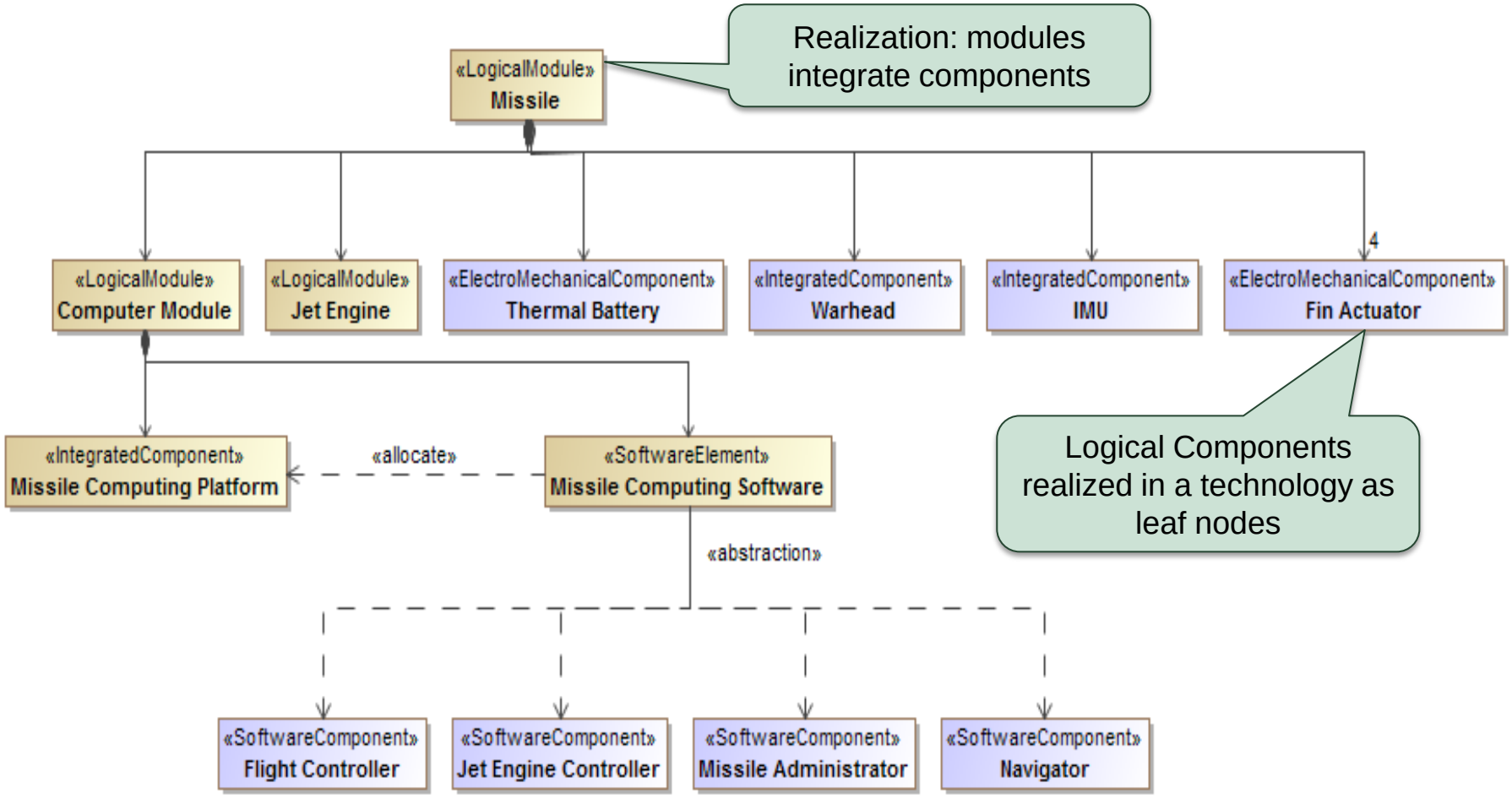




State and Activity Diagrams shows external visible behavior for System and Components



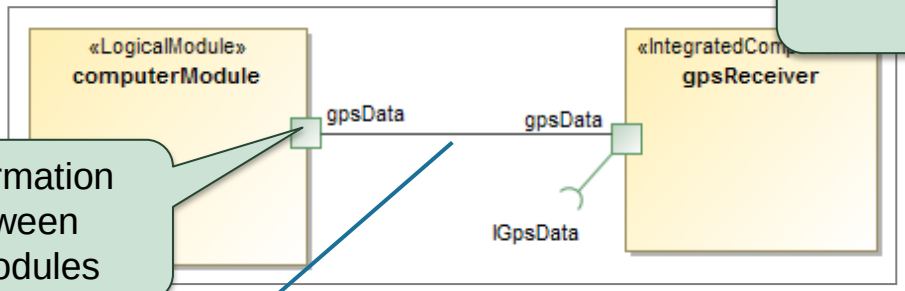
Missile – Logical architecture structure



Example from Missile Reference Model

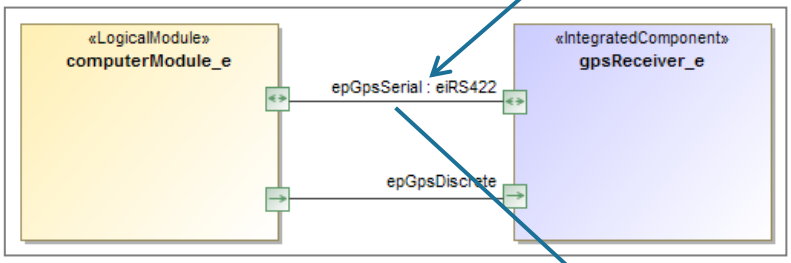
Logical Architecture – layers

Logical modules my have different layers

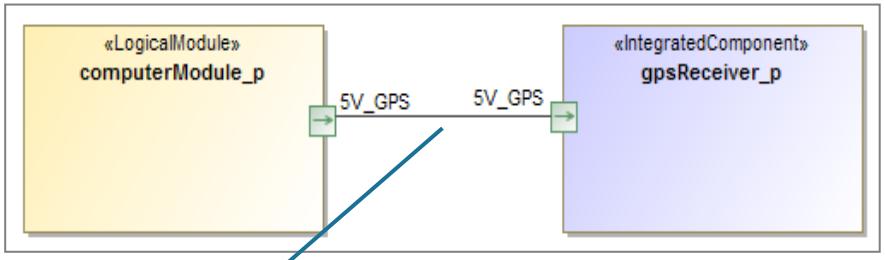


Information view

Defines the information exchange between components/modules

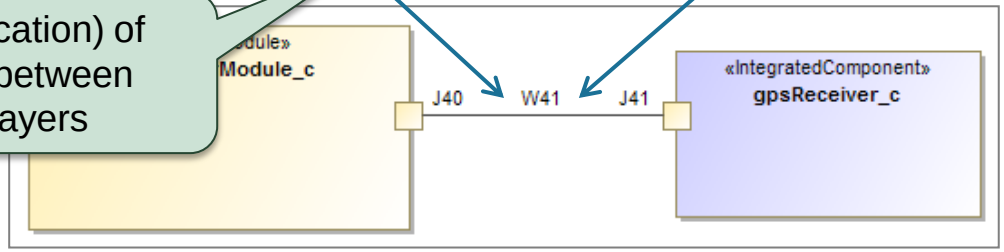


Electrical Signal view



Power distribution view

Tracing (allocation) of connectors between different layers

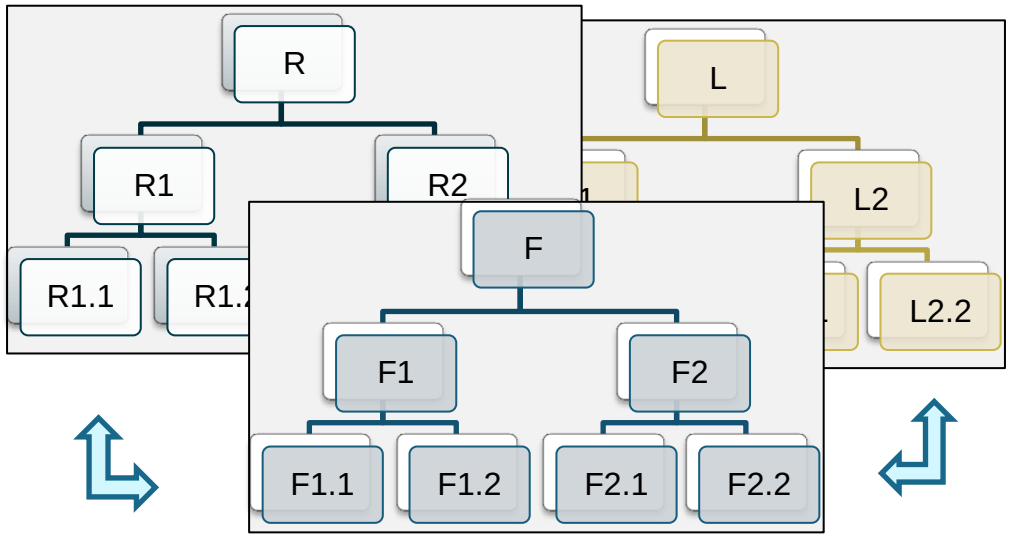


Cabling view

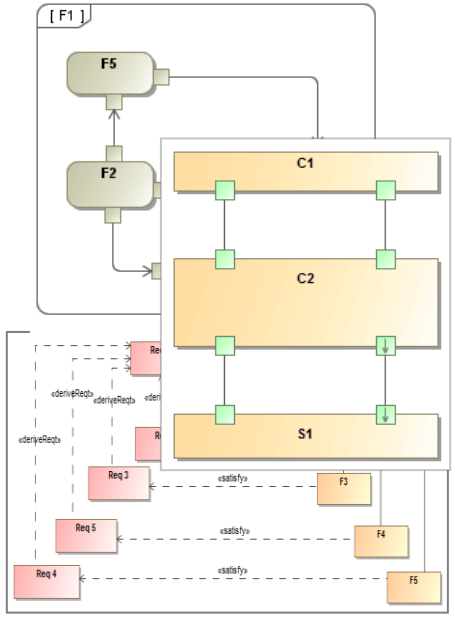
JSM System Model – summary

KONGSBERG

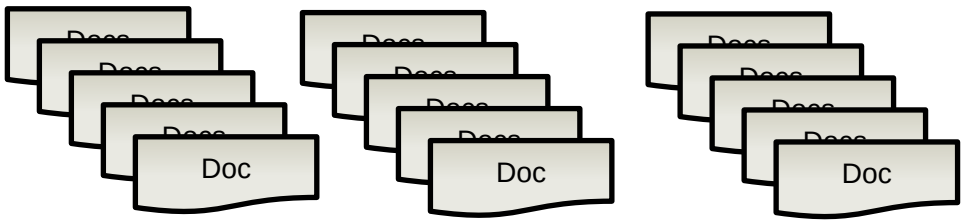
System Architecture Model

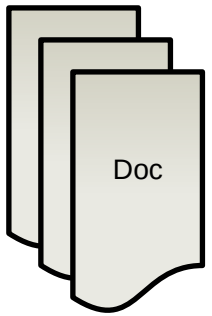


Diagrams



Generate





Is this a success story?

We have made a good foundation

- Established SAM expressing R,F, L and P of the JSM
  - > 25 systems, > 80 components, > 4000 diagrams
  - 20-30 persons contributed to modeling the SAM (R,F,L)
- Precise specifications for component development, especially for SW
  - Smooth transition to SW component design
  - Generating code for interfaces defined in SAM
- Commitment from Management

Success! But we still need to improve and evolve…...

- integration of the next JSM product increments must be successful
- «everyone» has to understand the model
- new employees should efficiently maintain the product
- model (elements) should be reused from JSM in other product variants
- the modeling culture must be sustainable

Lessons learned

Experiences and recommendations

KONGSBERG

Experiences and Recommendations #1

- Adopting a MBSE solution is a long journey
  - It's about learning new methodology, new architecture framework and a new language/tool in parallel
    - In JSM it took several years to get the Architecture Framework mature
      - 10 Workshops and trainings (2-3 days), extensive mentoring
      - MBSE test bed – tested in the *Local Hawk* student project
  - Invest in training and mentoring, Establish core team(s) and mentor(s)
- Keeping the model update and consistent is mandatory
  - Do not put too much details into the model
  - Throw away duplicated/obsoleted information
- SysML very expressive and powerful, but complex
  - Define a language subset and a strict guideline to develop large models -> Establish a reference model expressing which subset of sysML to use for which purpose

Experiences and Recommendations #2

- Systems Engineering terminology is overloaded
  - Functional versus logical? What is a system?
  - Clear terminology is essential in communicating the model -> define!
- Well managed abstractions is required to manage complexity
  - abstractions are not popular at the first glance for many
    - «abstractions hiding the details that is important»
    - «the information become fragmented by applying separate views»
  - Defining good abstractions is hard work, but it is worth the effort!
- It is a challenge to develop methodology and guidelines in parallel with product development
  - Start small: Establish methodology and Architecture Framework on pilot projects or small products/small parts of a product
  - Documenting existing products components – good way to learn and establish methodology & framework, «sandwich» process – meet in the middle
  - Roll out stuff that works!

The recipe for success

think **BIG**

start **SMALL**

and **EVOLVE**

Questions?