The Big Happy Family of System Architecture Approaches

Chris Phillips
14 Jun 2018

Agenda

- Introduction
- Overview
- Key Definitions
- System Architecture Overview
- Architectural Approaches
- Integrating Architectural Approaches
- Conclusion

Obligatory Briefing Start Cartoon & Quote

- "All architecture is great architecture after sunset; perhaps architecture is really a nocturnal art, like the art of fireworks." – G.K. Chesterton

Source: https://www.gocomics.com/calvinandhobbes/2012/05/24/

Introduction

- Systems Engineer – current focus on System Architecting
- Education:
  - BS – Engineering (Electrical Concentration / Specialty) – Colorado School of Mines - 2007
  - MS – Applied Systems Engineering – Georgia Institute of Technology – 2017
- Member of Coast Guard Auxiliary

Overview

- Large focus within Systems Engineering on the development and use of descriptive modeling tools, methods, techniques, etc.
- Descriptive modeling typically focuses on descriptive modeling of a system's architecture
- Various paradigms / approaches have been developed for describing system architectures
  - Each approach is suited for a different purpose
- For large, complex systems, integrating multiple approaches is typically required
- Purpose of this presentation is to introduce main approaches and discuss methods for integrating them

Key Definitions / Terms

- Architecture – the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution [1, 2]
- Architecture Description – a collection of artifacts or work products used to describe an architecture [3]
- Architecture Framework – describes the principles and practices used to develop an architecture description
- Architecture Model – a representation of a model which typically consists of numerous constituent models including descriptive models, analytical models, requirements models, etc.
- Metamodel – "Model of the model". Describes the conventions, relationships, etc. used within an architecture model

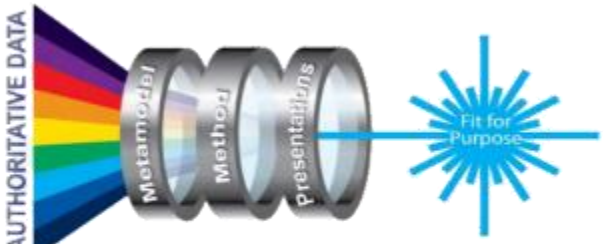
[1]: ANSI/IEEE 1471-2000
[2]: ISO/IEC 42010:2007
[3]: IEEE 1471-2007 Conceptual Framework

System Architecture Overview

- "Every System has [at least one] architecture" [1]
  - True whether documented or not
- An architectural description includes: [1]
  - Identification of stakeholders
  - Architectural concerns
  - Architectural viewpoints
  - Architectural views
  - Architectural models
- NOTE: Architecture, Architectural Description, and Architecture Model often interchangeable (especially if using a Model-Based approach)
- [1]: IEEE 1471-2007 Conceptual Framework for Architectural Description

Architectural Approaches

- Numerous approaches exist:
  - Enterprise
  - Service-oriented
  - Solution-oriented
  - Product-line
  - IT System
  - Etc.



DoDAF V2.0

ZACHMAN INTERNATIONAL®
ENTERPRISE ARCHITECTURE

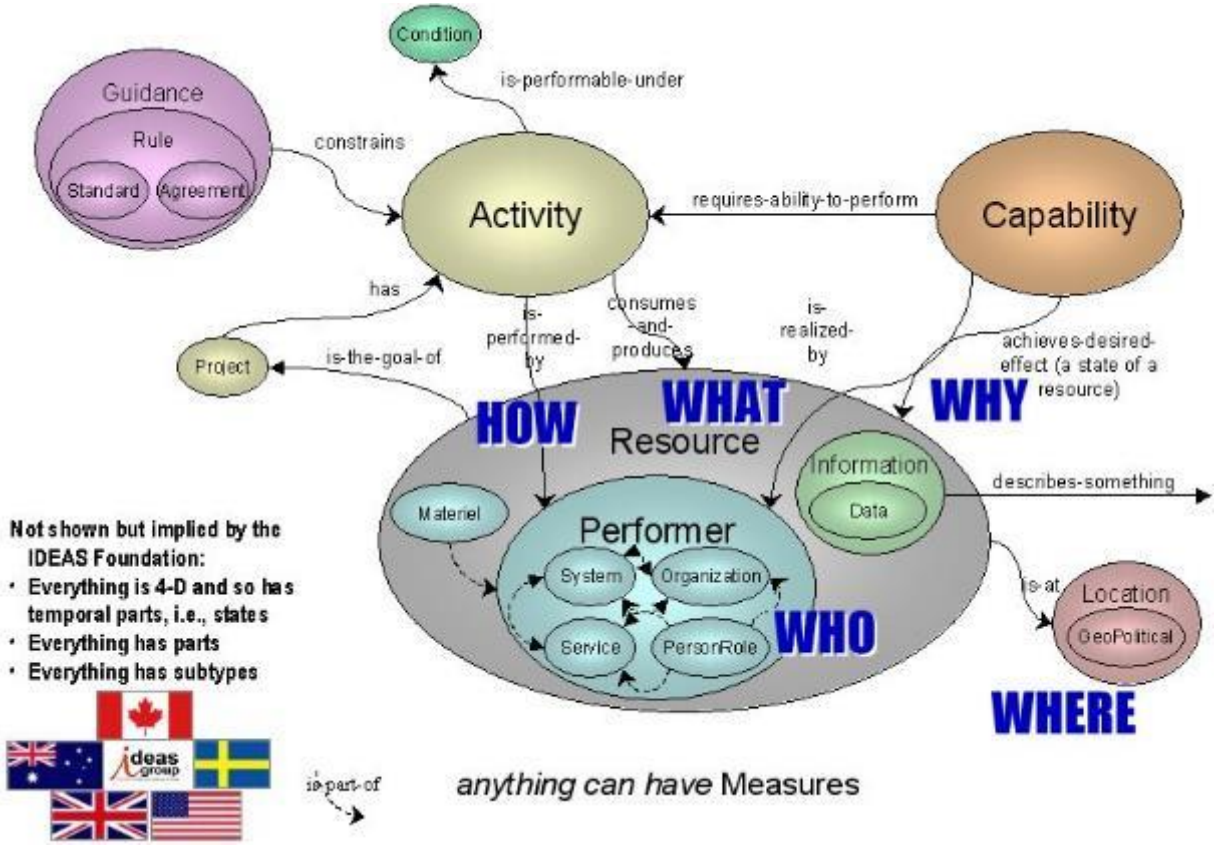
OMG SYSTEMS MODELING LANGUAGE™

SOAML

Approach 1: Enterprise Architecture

- Description: "well-defined practice for conducting enterprise analysis, design, planning, and implementation, using a holistic approach at all times, for the successful development and execution of strategy." [1]
- Highly abstract / conceptual. Describes system elements in terms of provided capabilities and use within an operational context
- Useful for integrating large system-of-systems especially within broader federation of systems
- Examples: DoDAF, MoDAF

Approach 1 Example: Universal Core

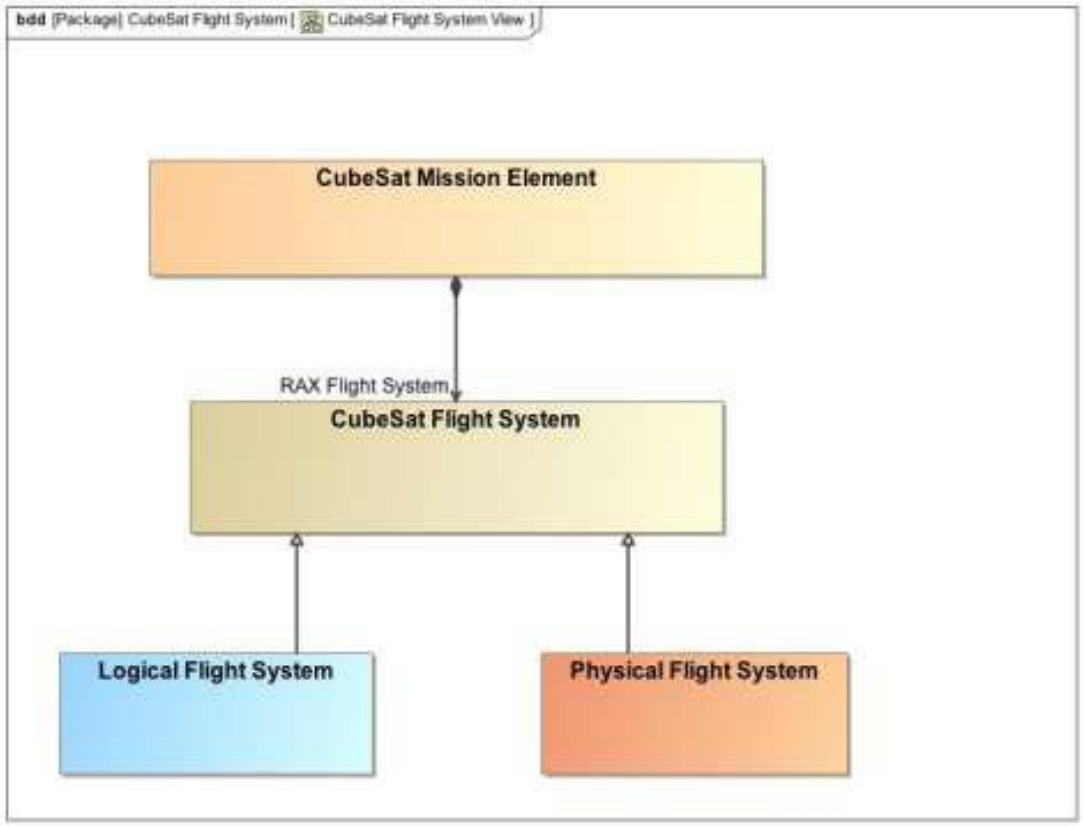


Source: https://dodcio.defense.gov/Library/DoD-Architecture-Framework/dodaf20_conceptual2/

Approach 2: Solution-Oriented Architecture

- Description: considered the "typical" architecture for a system designed to meet a particular need. Easily mapped to the SE V Model. Describes system from perspectives of requirements, functionality, and / or structure.
- Highly tailorable to address multiple levels of abstraction in all three domains
  - Example: Conceptual (Use Cases / Operational) -> Logical (Desired Functionality) -> Physical (Actual Functionality)
- Useful for describing standalone systems.
- Numerous examples

Approach 2 Example: Basic CubeSat Flight System Framework



Source: http://www.omgsysml.org/mbse_cubesat_v1-2012_ieee_aero_confr.pdf

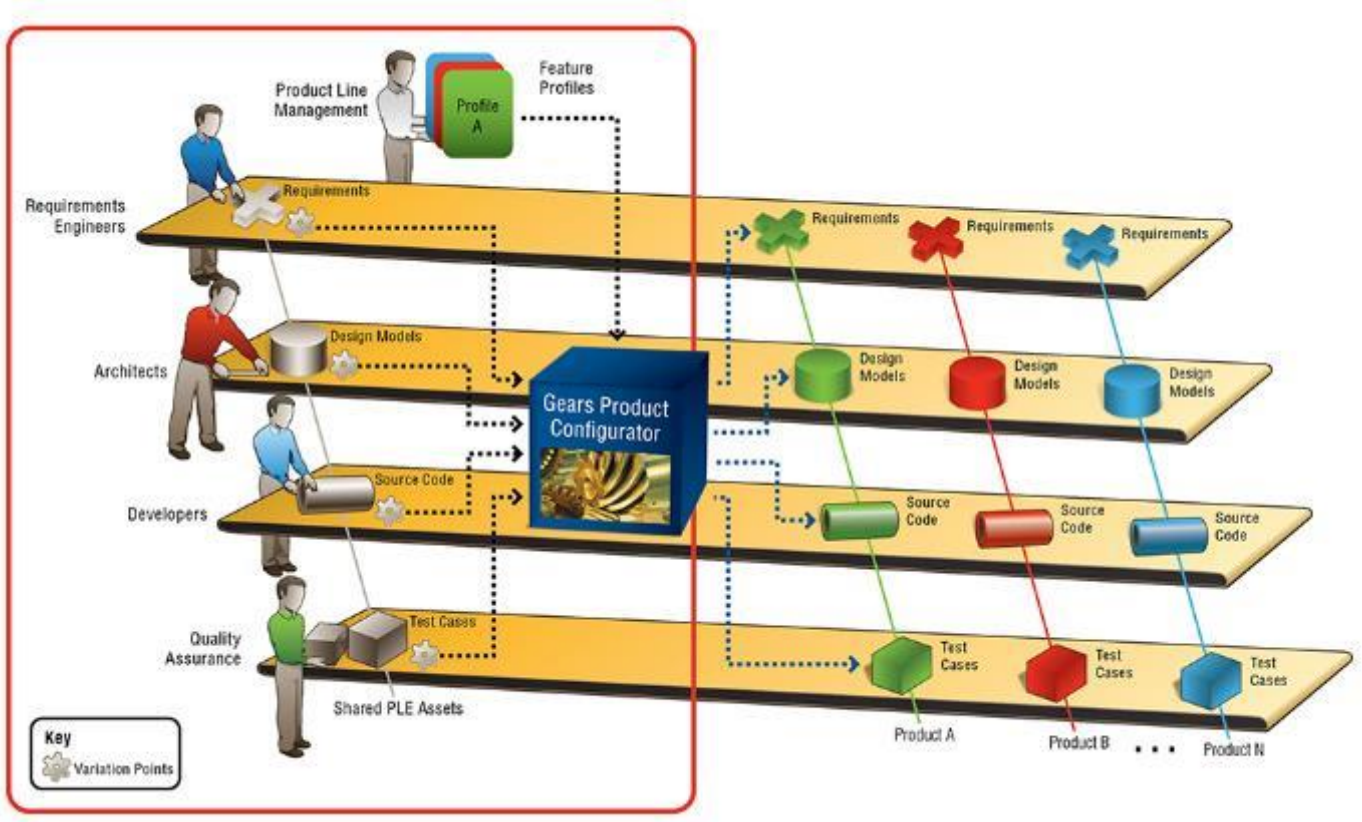
Approach 3: Service-Oriented Architecture

- Description: A set of components which can be invoked, and whose interface descriptions can be published and discovered [World Wide Web Consortium]
- Also called net-centric. Treats individual components as black boxes that execute functions / provide data & services
- Most typically used for software-intensive systems w/ strong object-oriented design. Can also be used for hardware-oriented system-of-systems especially if kept at conceptual / abstract level.
- Example: World Wide Web

Approach 4: Product-line Architecture

- Description: Describes a product model or series of product models based on the desire to provide a generically applicable solution or set of solutions to a range of problems.
- Typically very concrete – focus is on describing a solution for use in other models which may be more abstract
- Architecture serves same / similar purpose as data sheet
- Relatively new approach – still maturing practices & techniques
- Examples: Automobile, COTS equipment

Product-Line Approach: Vision

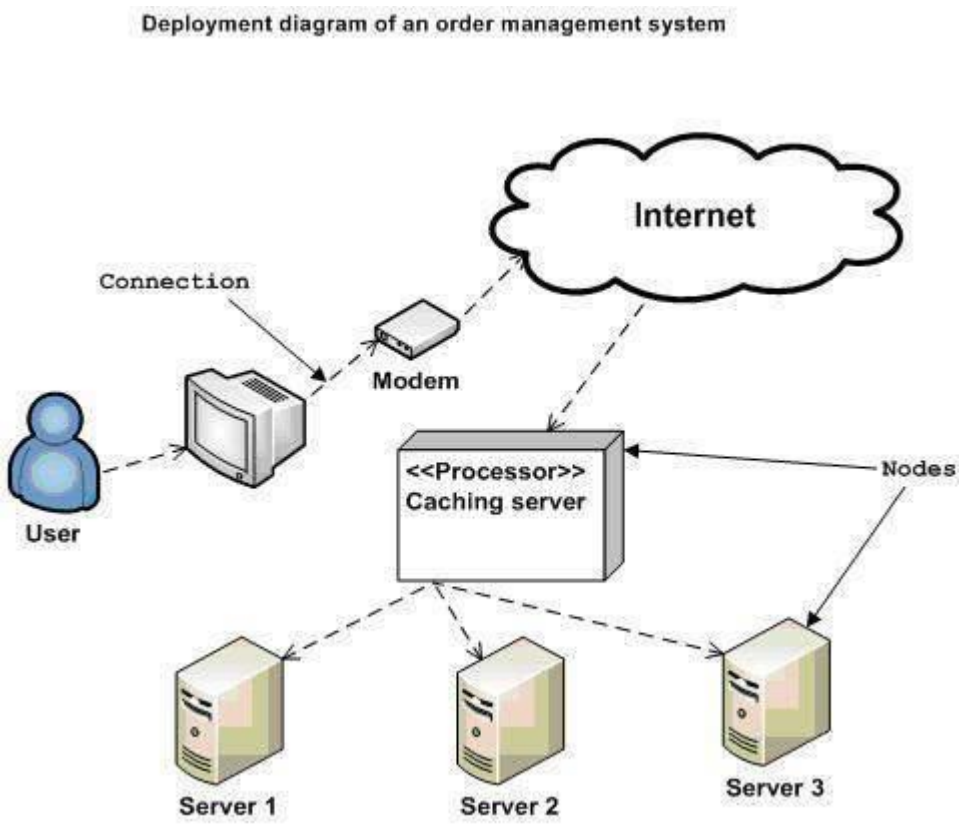


Source: http://www.productlineengineering.com/concepts/ple-defined.html

Approach 5: IT Architecture

- Description: describes a set of resources that will be deployed to provide a required set of capabilities
- Combines aspects of other approaches as required.
- Treats software & communications as primary interface mechanism. Hardware components and interfaces typically considered peripheral
- Examples: Network deployment diagrams

Example: IT Architecture



Architecture Approaches - Selection

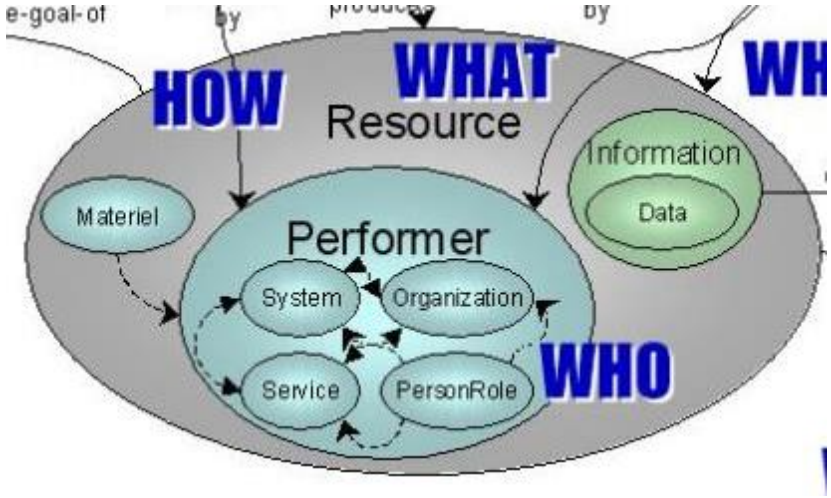
- Each project / program and organization typically selects an overall approach at inception
- If required, overall framework is setup / implemented
  - Can be done at start or as time progresses
  - Ultimate result is Architecture Framework and Metamodel
- Choice driven by various factors:
  - Contractual Requirements
  - Purpose of the Architecture
  - Business Model
  - Personal Preferences
  - Best Practices (internal or external)
  - Architectural characteristics (Standalone or SoS, complexity, complication, etc.)

Integrating Architecture Approaches

- Goal of MBSE: use models to describe and understand systems
  - Complex systems typically utilize federation of models within MBSE approach
  - Descriptive (Architecture) Models within federation may use numerous approaches & styles
- Challenge is to federate model types. Typical examples include:
  - Government / Contractor (Enterprise + Something else)
  - IoT Design Agent (Solution-oriented + Service-oriented [WWW])
  - System designer vs. potential suppliers (Solution-oriented + Product-Line)

Integrating Architecture Approaches – Enterprise with Everything

- Enterprise Architecture goal is to be an integration point
- Enterprise Architecture Frameworks have defined level of high abstraction
  - Individual systems (people, products, etc.) represented as black box
  - Lower-level details can be detailed out in separate architecture descriptions



Integrating Architecture Approaches – Solution-Oriented with Service-Oriented

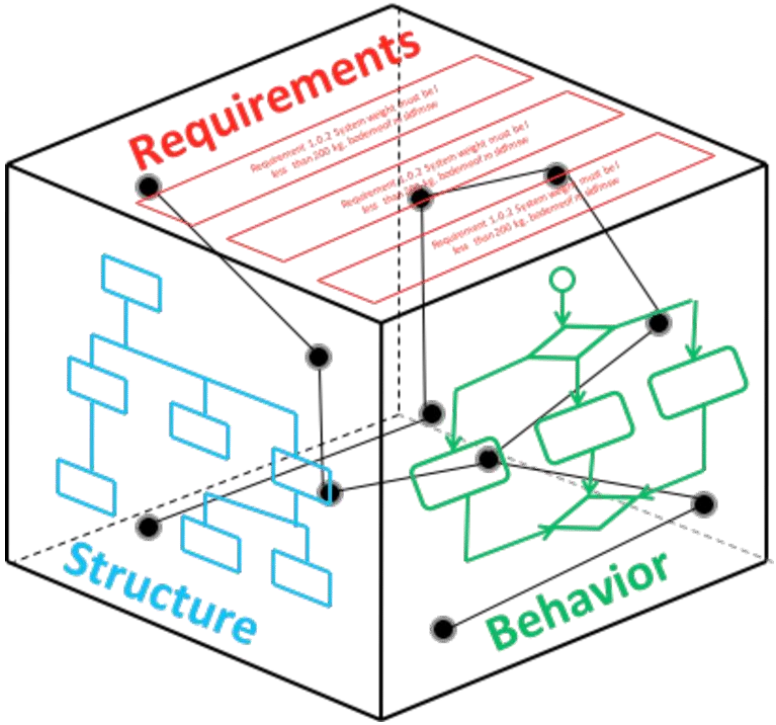
- Typical approach is to establish top Solution-oriented item up as a Service-providing element
  - Leverages Service-Oriented "Black Box" concepts (similar to Enterprise)
  - Typically easier to detail out internal elements using the solution-oriented approach
    - Can be difficult / burdensome to maintain largely redundant parallel architectures
  - Can be leveraged for key requirements & system definition processes:
    - Functional / Use Case Analysis
    - External Interface Definition
  - Allows for invocation of behavior from external actors

Integrating Architecture Approaches – Solution-oriented with Product-line

- Area that will need to be addressed soon as Product-line Architecture Concepts mature
  - Challenges arise when performing trade studies, during sustainment / replacement projects, etc.
- Numerous factors
  - Different internal approaches
  - Limited and loose community standards
  - Different goals of each architecture
  - Proprietary / sensitive data (both sides)
- Two main approaches seen to date (not mutually exclusive):
  - Leveraging Domain Cross-cutting relationships
  - Heavy use of Specializations

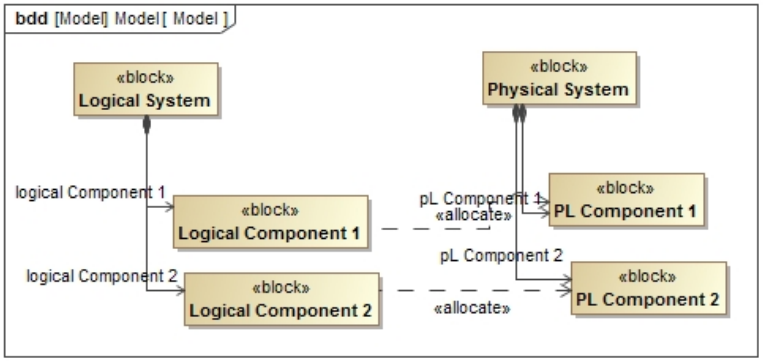
Solution and Product-line Approaches: Integrating with Domain Cross-Cutting Relationships

- Cross-cutting relationships primarily correlate Requirements, Structure, and Behavior
- Considered relatively weak (shown as dashed line in SysML)
- Also used to tie sub-tier elements of single domain (e.g., Logical & Physical)



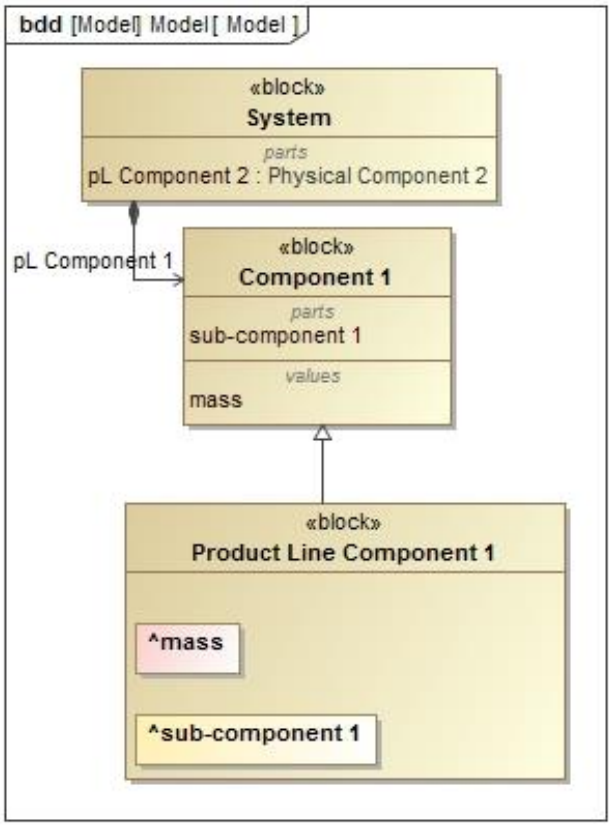
Solution and Product-line Approaches: Integrating with Domain Cross-Cutting Relationships

- Example implementation: use of logical & physical domains
- Logical follows solution-oriented principles
  - Lower-level items considered definitional / requirements
- Physical uses product line items as though solution-oriented



Solution and Product-line Approaches: Specialization

- A generic representation is present within all domains
- Elements within the Product Line Architecture are created as specializations of the generic element
  - Required to provide same basic set of descriptive properties



Solution with Product-Line Architectures: Pros & Cons

Cross-Cutting

- Pros:
  - Integration across high-priority domains / areas
  - Products can be treated as static items in original state across multiple solutions
  - Easily understood separation of data between generators / owners
- Cons:
  - External interfaces (e.g., analytical models) may require additional wrappers built over time

Specializations

- Pros:
  - Continued Plug-and-play integration across model in all domains / areas
  - Creates apples-to-apples comparison mechanisms
- Cons:
  - Additional work to initially integrate
  - Risk of information overload (all information in one place)
  - Model maintenance activities may require maintenance of obsolete options
  - Product line architecture variants as each solution's architecture developed

Conclusion / Summary

- System Architecture continues to be combination of art and science
- As practice of System Architecting matures, various approaches may be used
- Approaches can be integrated in various ways depending on types & desired strength of relationships
- Integrating Product-line Architectures presents key set of challenges and opportunities


That's all Folks!