A Reusable Ground System Architecture Provides Lessons in the Use of DoDAF and UPDM

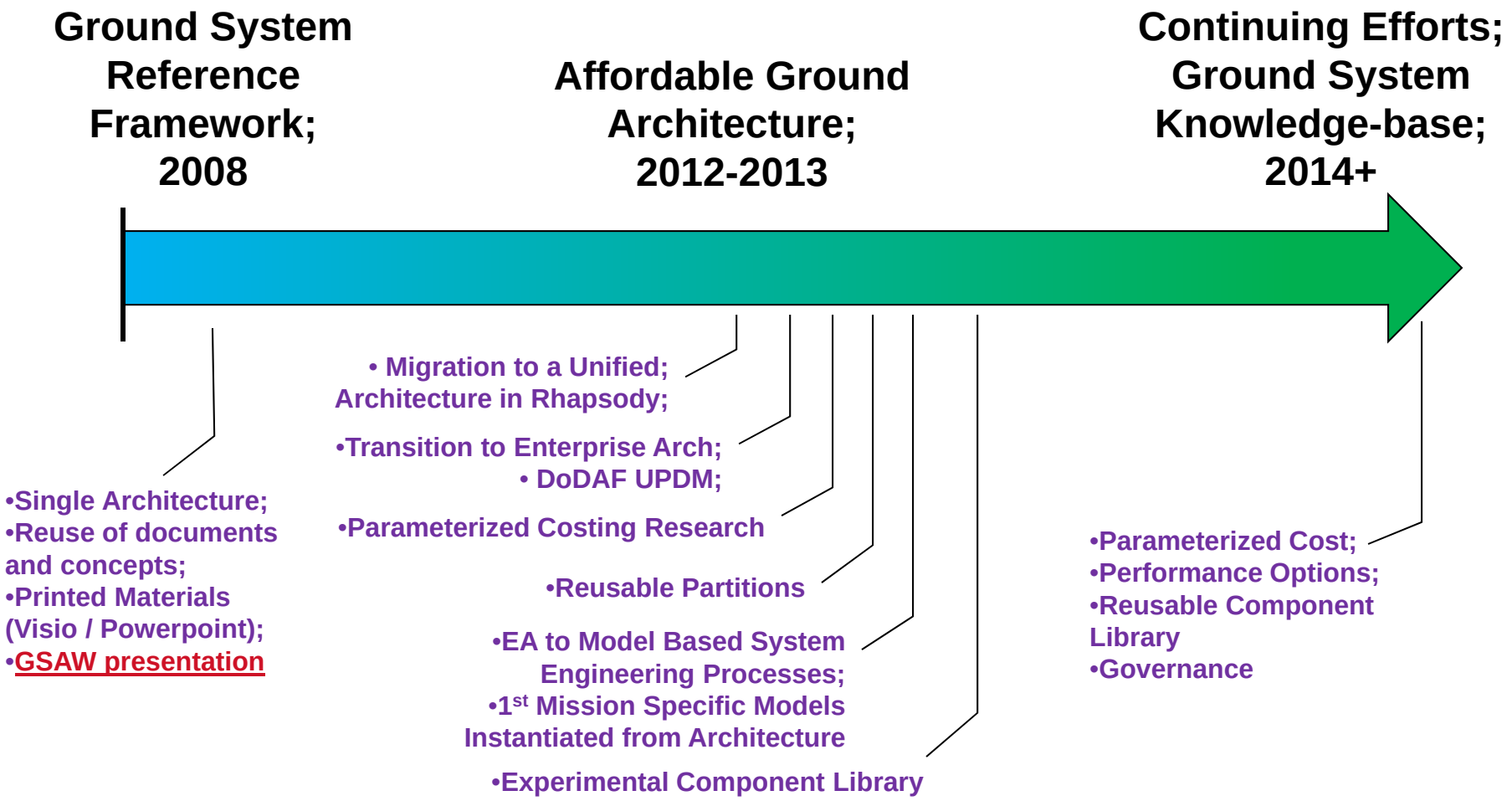
HARRIS®

Eric A. Barnhart
Systems Engineer
Howard D. Gans
Systems Engineer

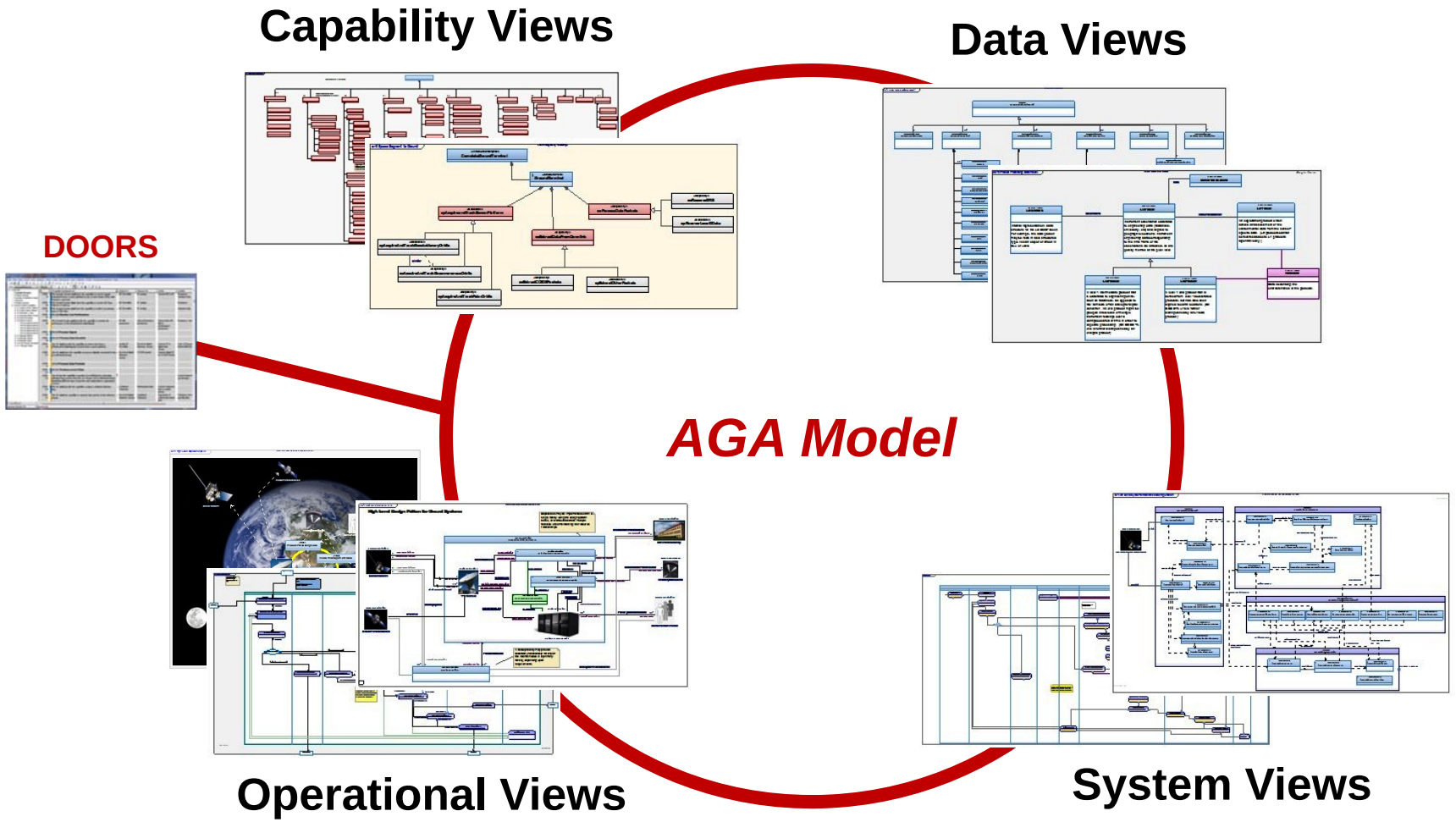
Background

- Affordable Ground Architecture (AGA)
  - A Harris initiative to develop reusable ground systems architectures applicable to multiple problem domains
  - Identifies problem-unique and common aspects for each domain
  - DoDAF 2.0
  - Unified Profile for DoDAF / MoDAF (UPDM) in Rational Rhapsody
- Based on experience in multiple ground systems
  - NOAA Geostationary Operational Environmental Satellite - R Series (GOES-R)
  - NASA Space Network Ground Segment Sustainment (SGSS)
  - U.S. Army's Modernization of Enterprise Terminals (MET)
  - 40 years operational experience at USAF/AFWA
- AGA Goals
  - reduce cost and execution risk through reuse

Timeline of the Harris Initiatives



AGA Overview



The AGA enterprise ground system model includes multiple DoDAF views

Issues in Architecture Reusability

Reusable

- Partitioning
- Naming conventions
- MBSE Transition Strategies

Multiple Enterprises

- Addresses Common & Unique Capabilities
  - Multiple arch. parts
  - Inheritance and classification

Multiple Decisions

- Capability Selection
- Component Libraries
- Parameterization
  - ✓Cost
  - ✓Performance

Non-reusable

- Simplified partitions
- No transitioning strategy needed

Single Enterprise

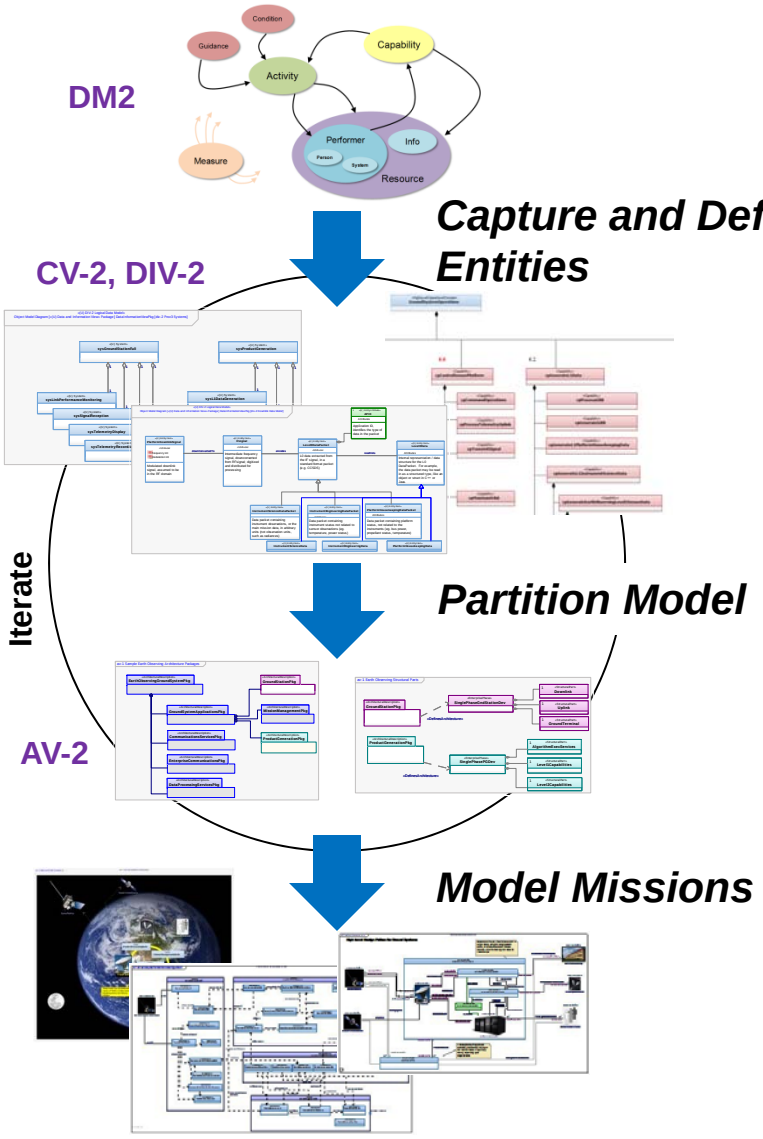
- Simplified Capability Model

Single Decision

- No parameterization needed
- No libraries needed

Reusability across multiple domains and decisions increases complexity

Architecting For Reusability

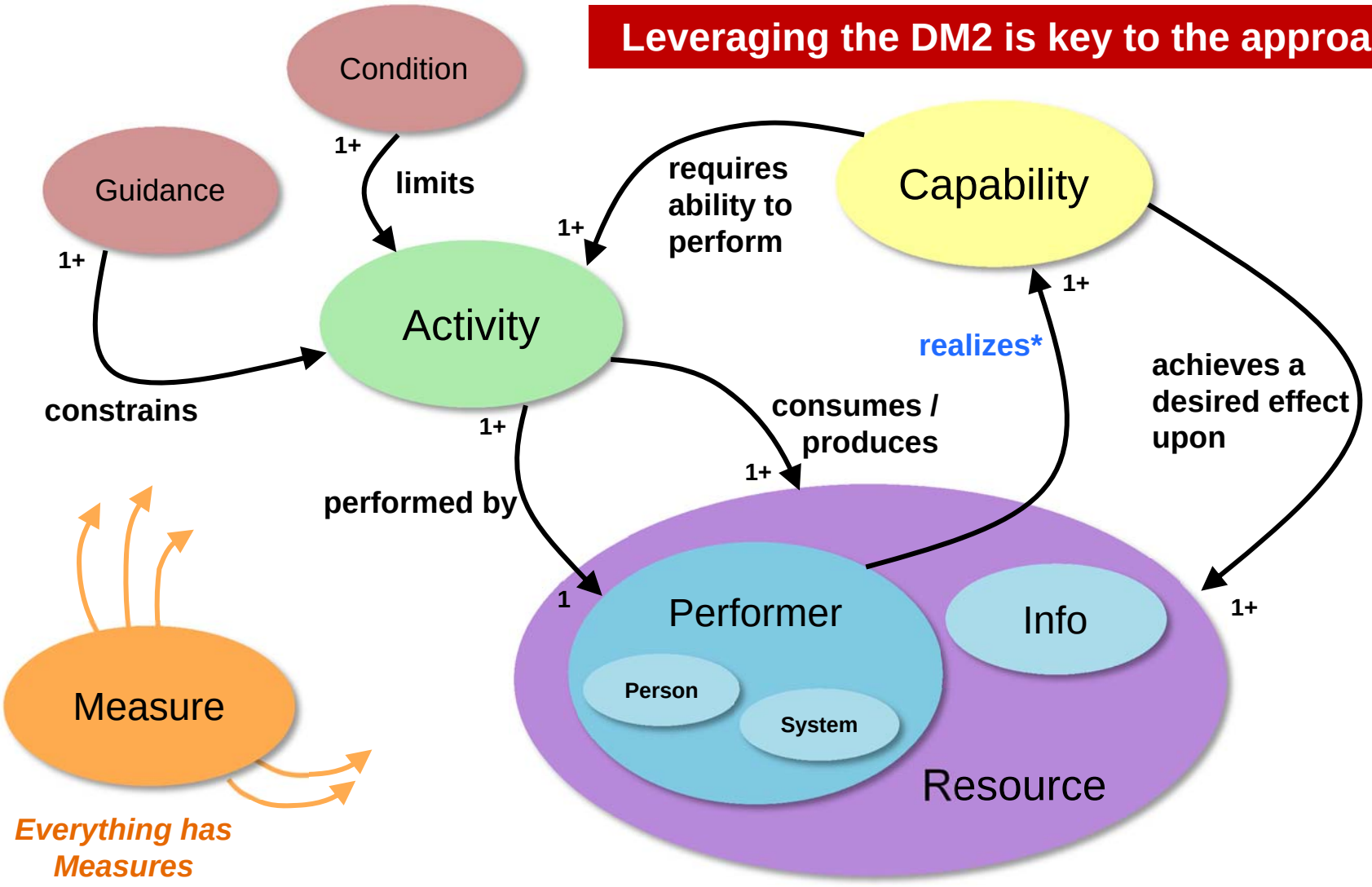


- First, model the ground system in terms of the DoDAF Metamodel (DM2)
  - Start with missions and capabilities (CV)
  - Defer the other views (OVs, SVs)
  - The other views fall out naturally after the meta model data are defined
- Define entities in a data view
  - Reusable entities need defined types
  - Use no implicit entities
  - DIV is equivalent to a Block Definition Diagram (BDD) in SysML
- Partition the architecture for reuse
- Use generic terminology to describe the EA entities

Experience suggests a modified approach to EA for reusability

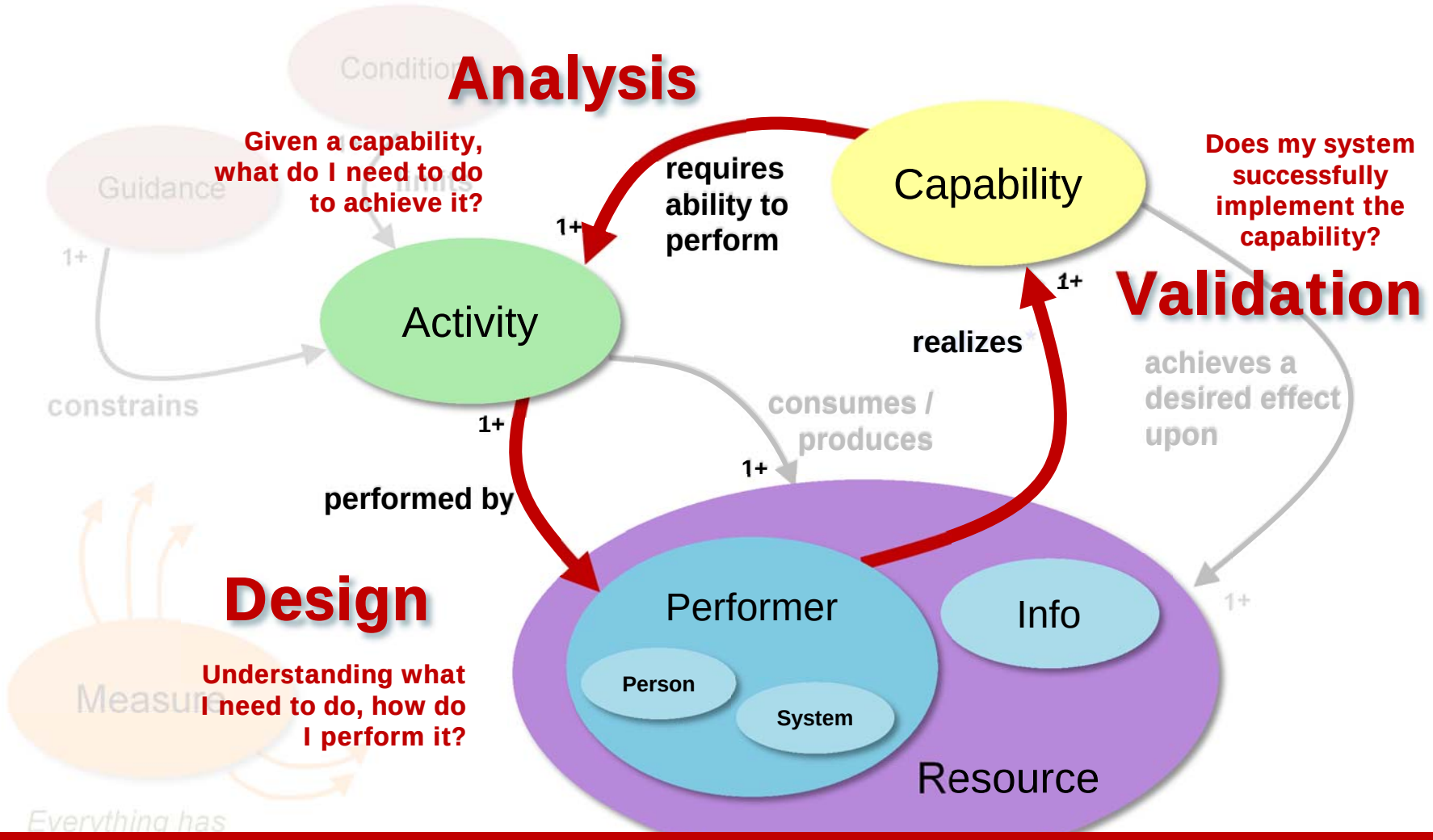
DoDAF MetaModel (DM2) Simplified

Leveraging the DM2 is key to the approach



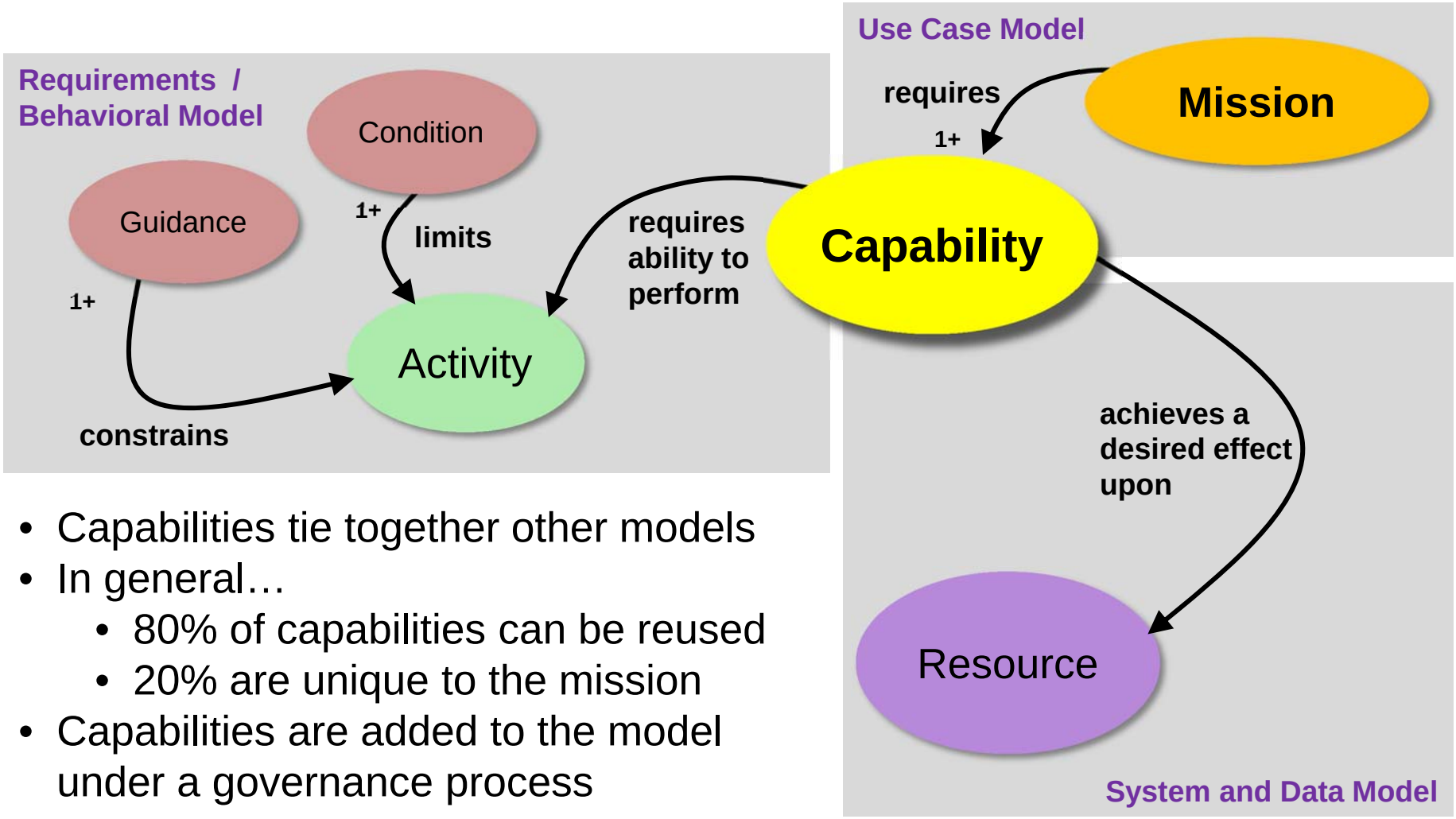
*relationship is inverted from the DM2

DM2 for Systems Engineers



The DM2 impacts the way systems engineers approach their processes

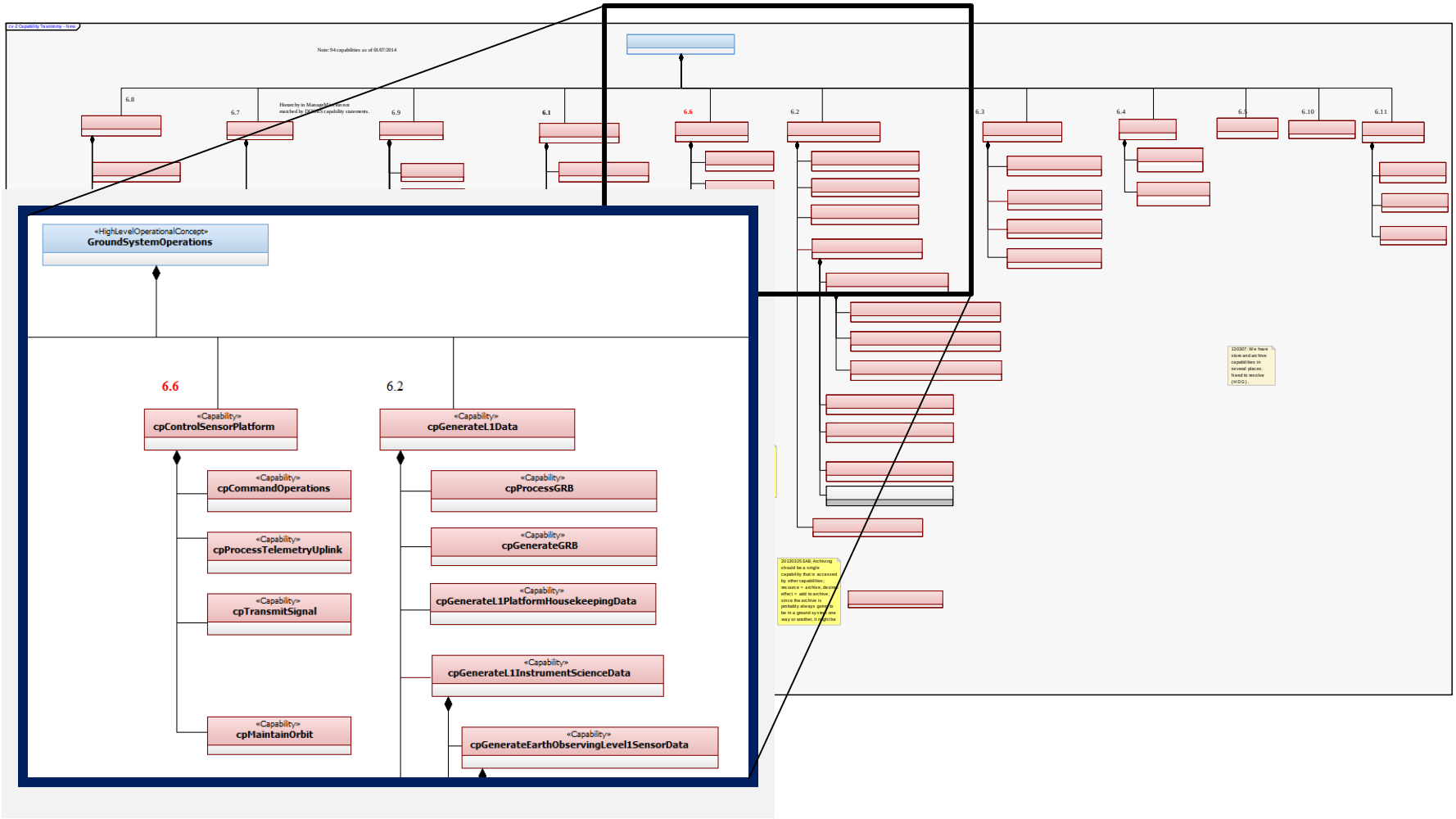
Capability Models



- Capabilities tie together other models
- In general…
  - 80% of capabilities can be reused
  - 20% are unique to the mission
- Capabilities are added to the model under a governance process

Reusable capabilities impact multiple systems engineering models

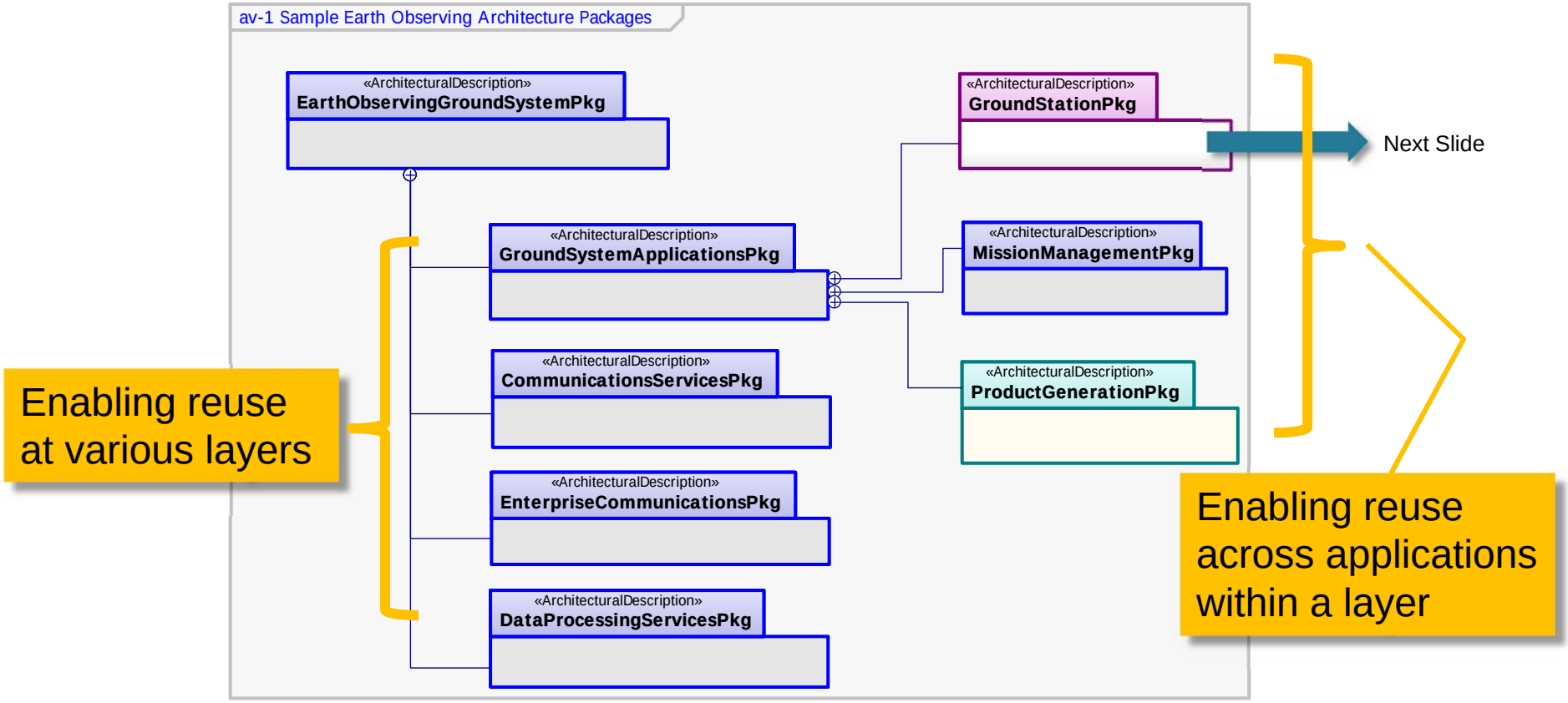
Capability Taxonomy



Hierarchical capabilities are useful for less complex, non-reusable architectures

Partitions Facilitate Modular Reuse

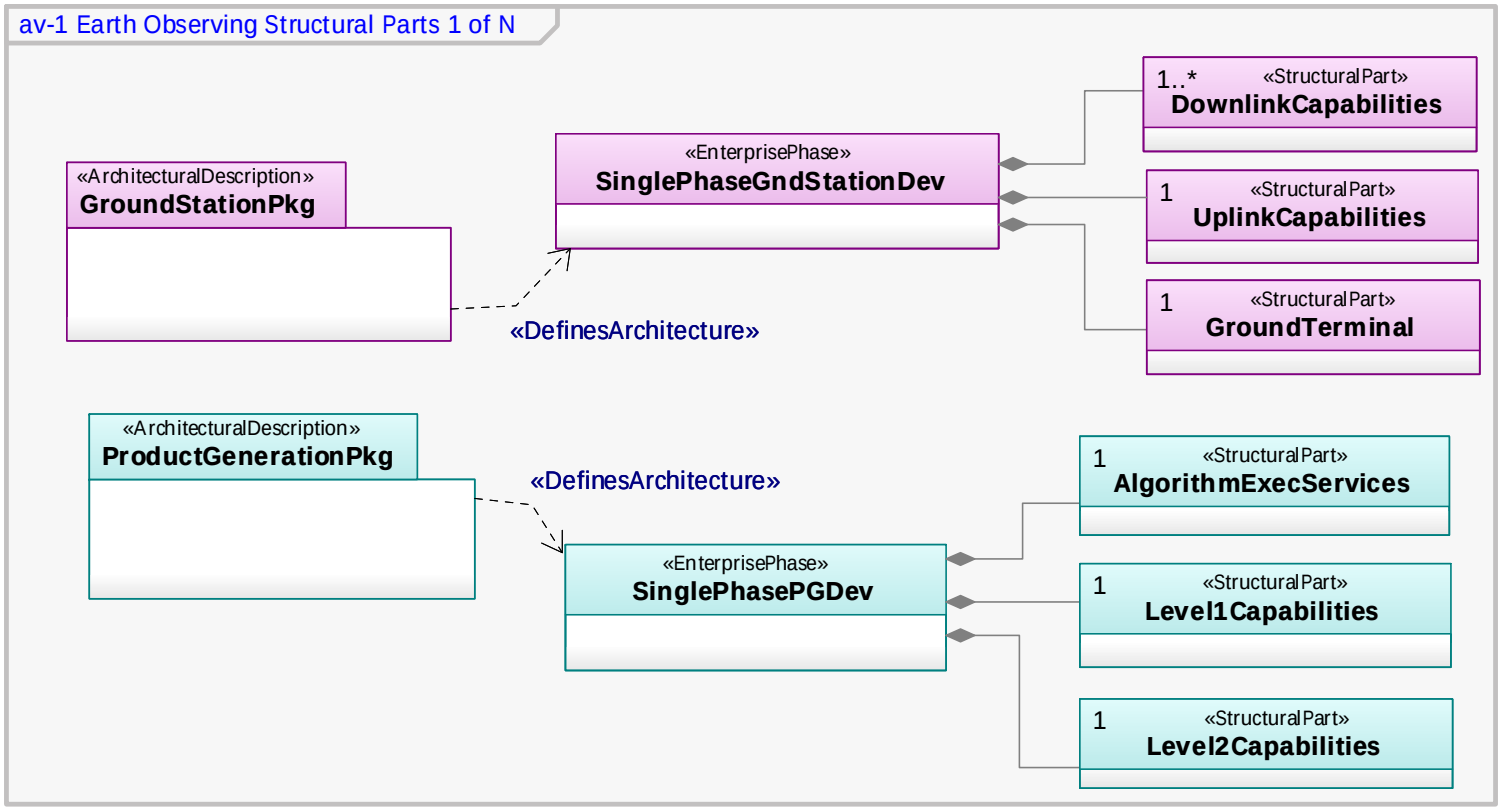
- Reuse is enhanced by an architecture partitioned into meaningful cohesive packages of capabilities



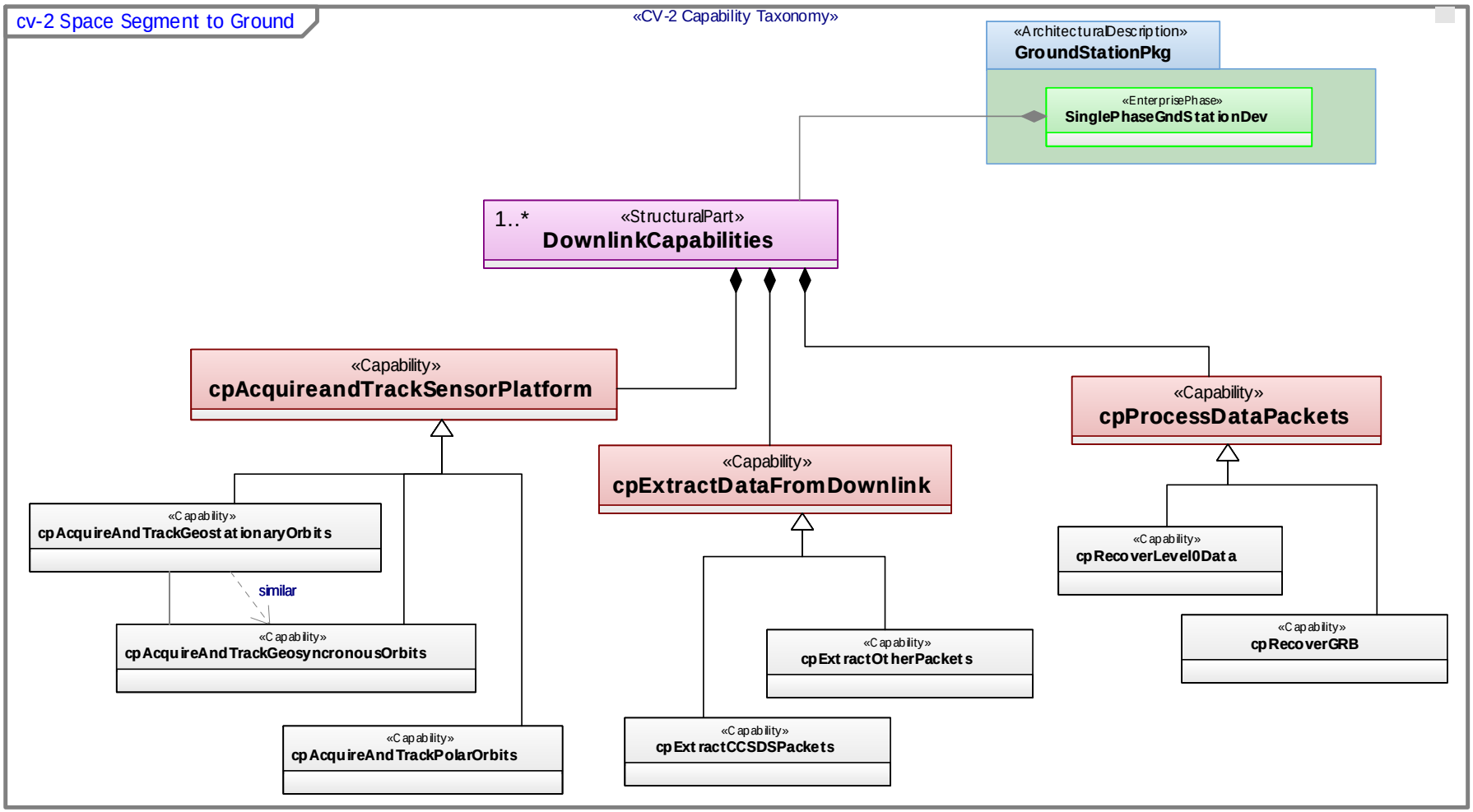
Architectural descriptions are imported into SysML models for reuse

Lower Level Partitioning

- UPDM provides a schema for grouping capabilities into temporal and/or structural parts within the architectural descriptions

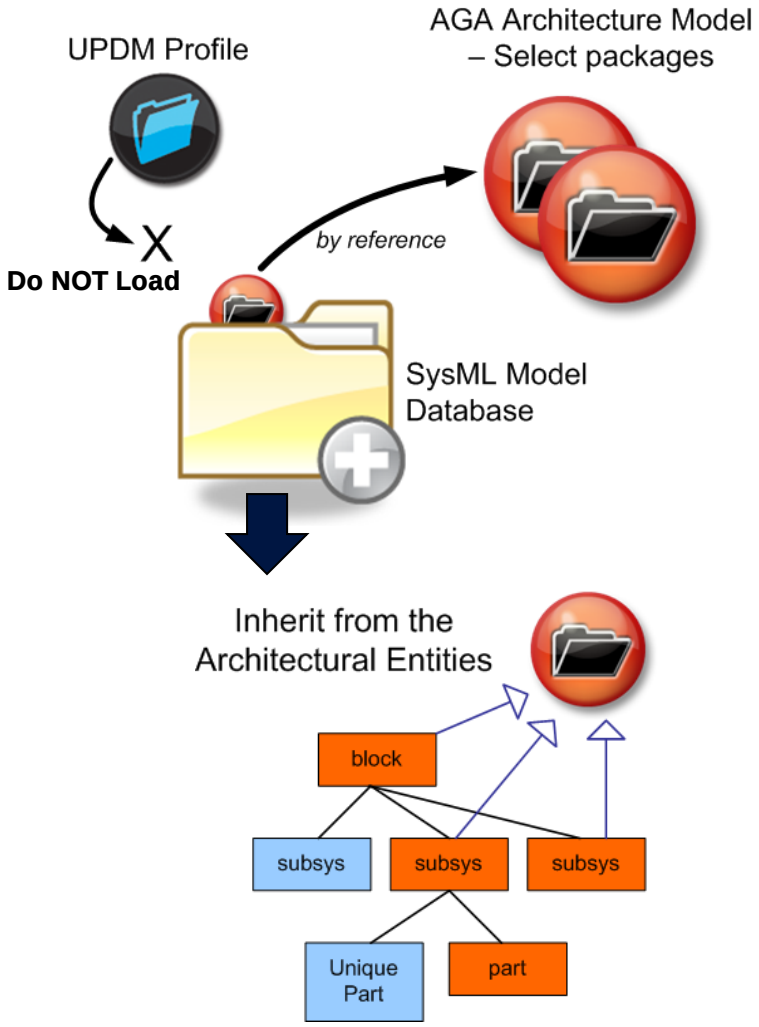


Enhanced Capability Taxonomy



Architectural descriptions facilitate reuse by grouping capabilities and their subtypes

Transition to MBSE / SysML



- Architectural Descriptions in UPDM are packages in SysML
- Transition to MBSE as follows
  - Import and reference these packages in a separate SysML project (readonly)
  - Do NOT load the UPDM profile!
    - Lacking the UPDM definitions, architectural entities now take on their equivalent SysML characteristics
  - Create mission specific SysML entities
    - Inherit the characteristics of the architectural entities
    - Inheritance does not require modify access
  - Customize the SysML model for the specific ground station mission

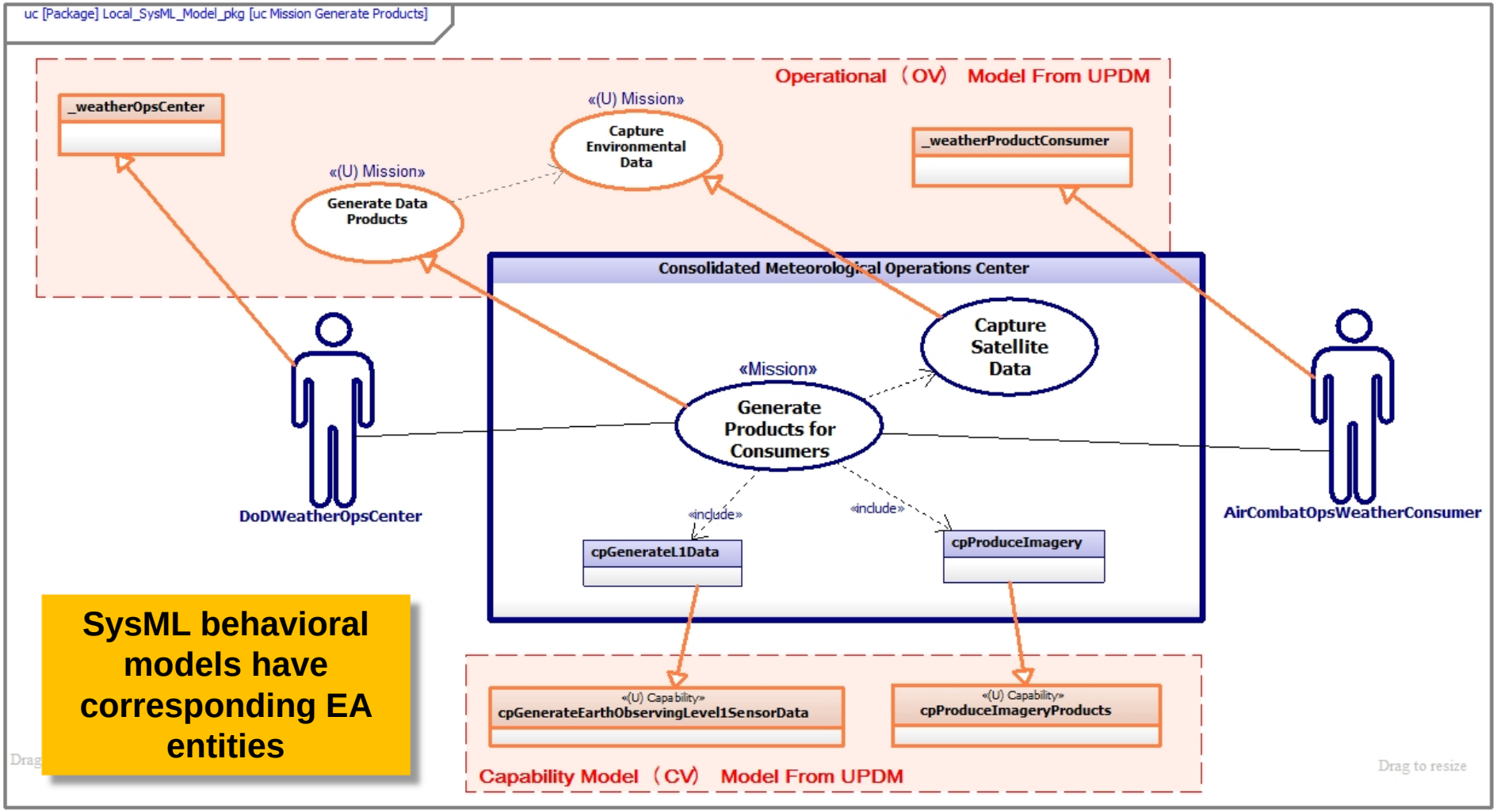
Architecture models transition easily to MBSE / SysML development

Generic Naming Conventions

- At the enterprise level, keep your taxonomy generic for reuse
  - Any facility where a weather product ground system is controlled can be called a “weatherCommandCenter” operational node
  - An orbiting sensor and satellite bus can be called an “orbitingObservatory” operational node
- At the system solution level, create specific instances
  - NOAA_WCDAS_Facility *is a type of* weatherCommandCenter
  - GOES-R_East *is a type of* orbitingObservatory
- For transitioning to SysML…
  - In order to map SysML system elements to EA elements, the EA elements must have a defined type
  - Implicit entities created directly on an SV-1 cannot be reused or mapped to SysML entities
    - First add them to a data view (DIV) as a defined entity

Generic enterprise taxonomies allow adaptation for unique missions

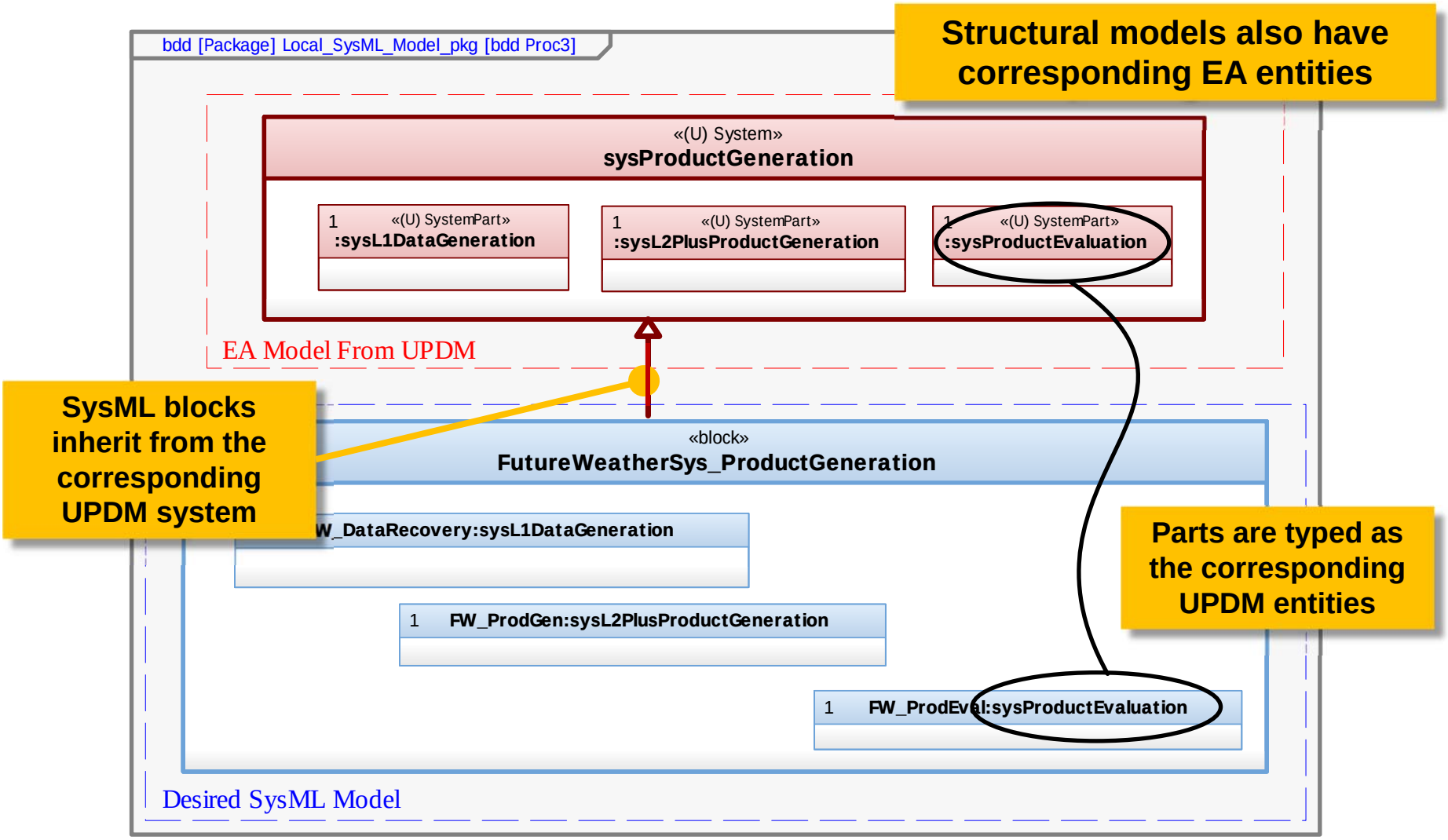
Example SysML UC Model



SysML behavioral models have corresponding EA entities

Underlying UPDM to SysML relationships enable this reuse process

Inheritance Of Architecture Characteristics



Summary

- Benefits
  - Enterprise Architecture can transition to Model Based Systems Engineering while maintaining traceability
    - Inherit the 80% common capabilities / systems
    - SysML can be used to engineer the 20% unique capabilities
  - SysML model is unique to the specific mission
    - Naming conventions and concepts are in mission terms
    - EA model is unmodified
    - EA model can be overridden where necessary

- Challenges
  - Manual process to make the links and unique configurations
  - Tools don't yet directly support this concept with built-in functions
    - Need to follow SysML relationships to identify inherited information
    - Opportunity for some scripting

Conclusions

- Modeling a Ground System Architecture for reuse has unique difficulties
- Implementing a ground system solution from a reusable architecture using SysML requires extensive knowledge of both UPDM and SysML
- Up front investment is not trivial
- The effort is worth the benefits
  - Consistent knowledge-base of ground system architectures
  - Reusability at several levels
    - Architectural components
    - System products
  - Rapid decision making capability
  - Lower Costs in up front processes
- Future benefits
  - Parametrically defined cost and performance
  - Reusable process artifacts
  - Reusable components