33rd Annual INCOSE international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023

Modeling System Configurations Over Time

Matthew Hause,
MHause@SystemXI.com

Lars-Olof Kihlström
Lars.Olof.Kihlstrom@cag.se

Agenda

- Background Information
- SysML/Object Orientation Issues
- Example Model
- Conclusions
- Questions?

The Evolution of Systems Over Time

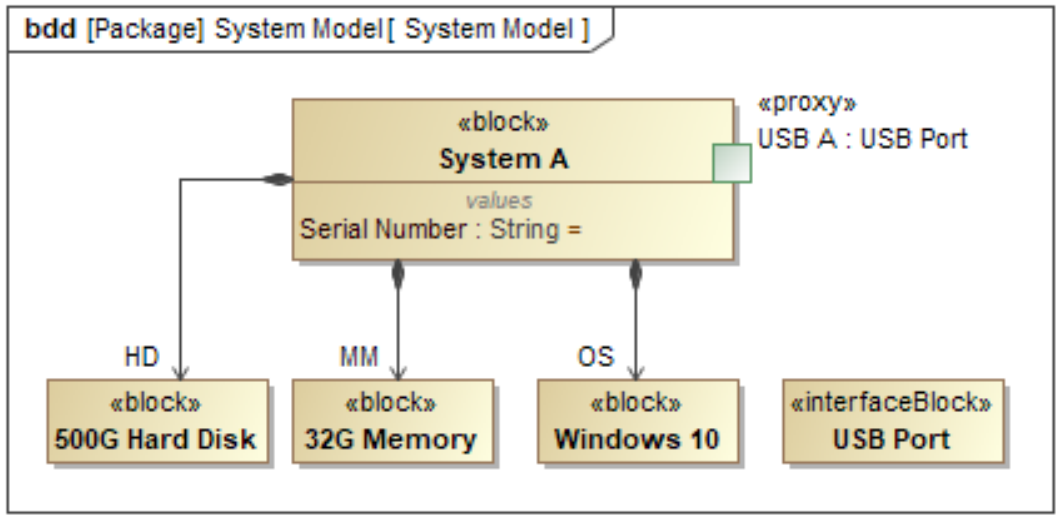
- Systems change over time in structure, functionality, capability, adaptability, etc.
- Systems can be configurable in bother behavior and structure
  - New software can be added to a smart phone
  - New hardware components and software applications can be installed on a laptop
  - More radical changes usually require a new platform

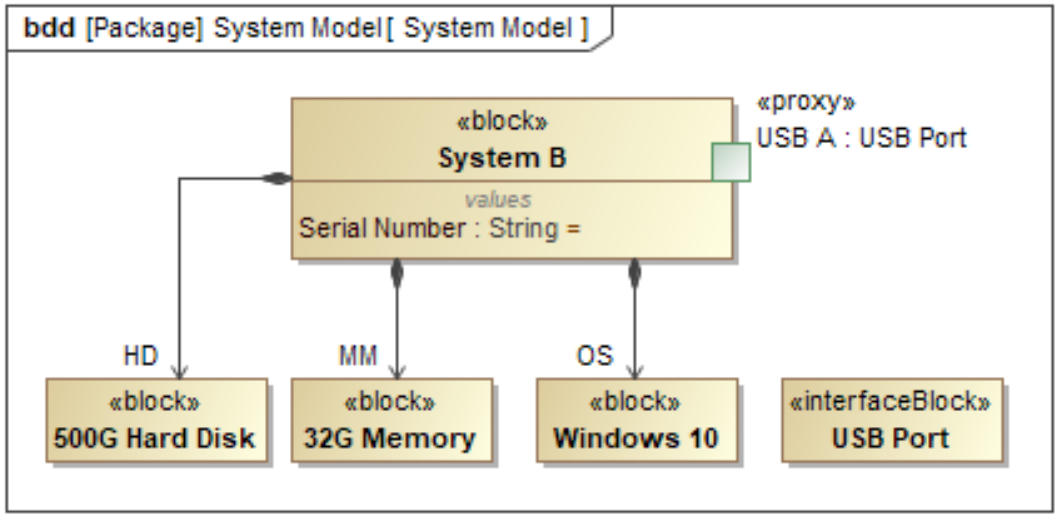
Exactly the Same, Only Different

- The paradox of the Ship of Theseus – Plutarch
  - If you take all the parts of a system (Theseus' ship) and replace them, is it the same ship?
- Abe Lincoln's axe
  - Lincoln was well known for his ability with an axe, and axes associated with his life are held in various museums.
  - Are they all "Abe Lincoln's Axe"?
- Systems change over time
  - System lifecycle of design, manufacture, deployment, maintenance, retirement
  - Changes for mission-based configurations
  - Changes due to maintenance
  - Etc.

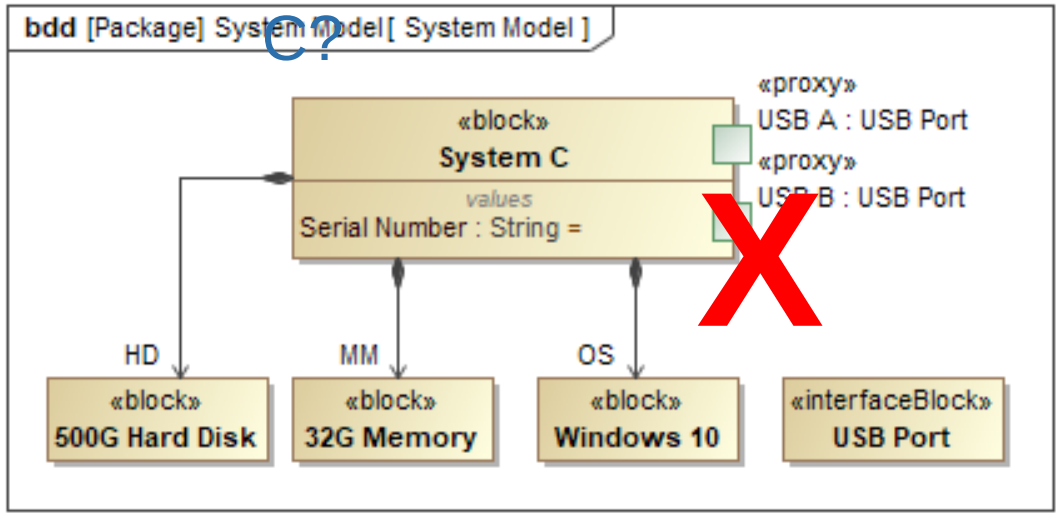
Modeling Systems in SysML

- Are System A and System B the same System?

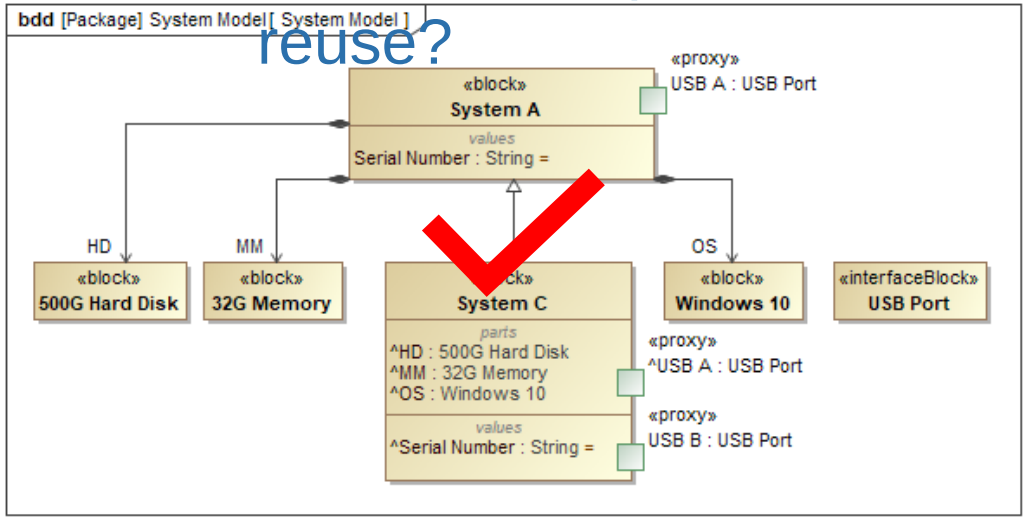




How about System C?



Does this help reuse?



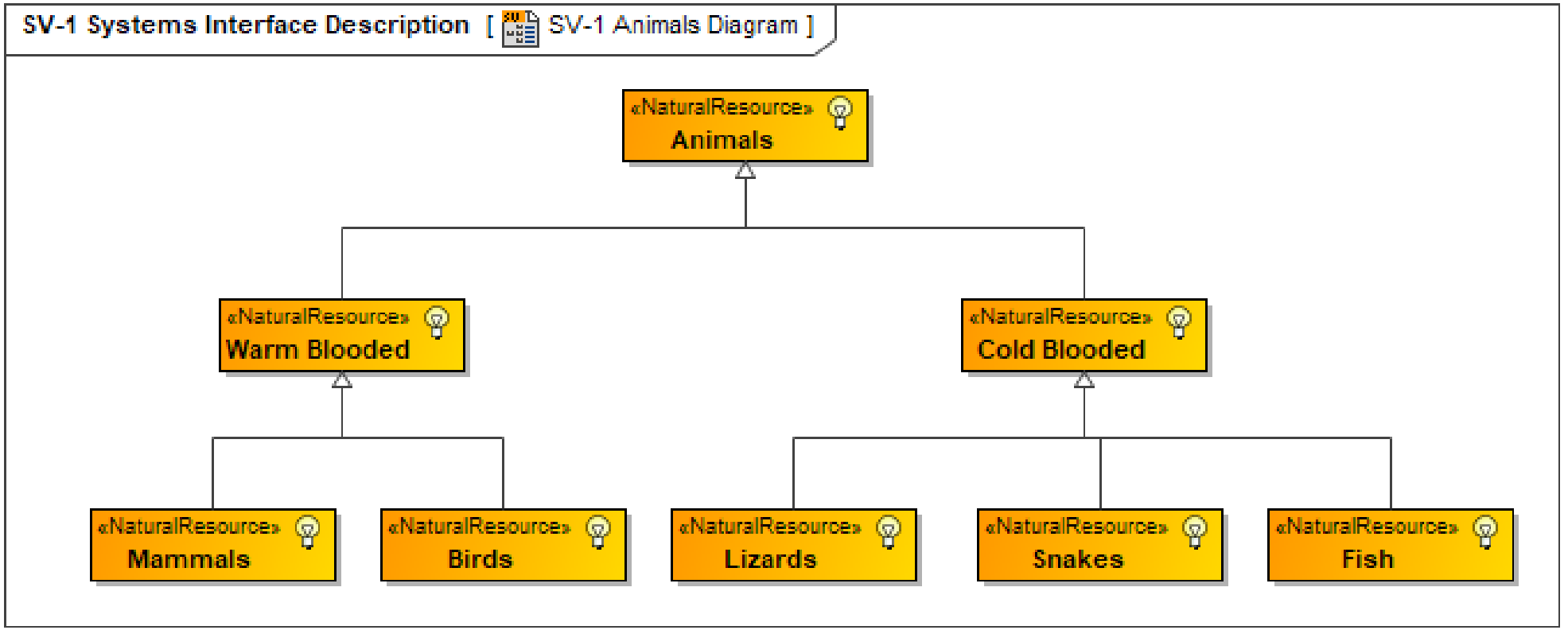
Inheritance in Object-Orientation

- Inheritance is the concept in which one class inherits the attributes, (properties & relationships) and methods of another class. The inherited class is the Parent class, the class that inherits the properties is the Child class.
- Along with the inherited properties and methods, the child class can have its own properties and methods.
- This allows us to define common characteristics as well as variations
- This can have both positive and negative effects.

https://www.deviantart.com/pheeph/art/ZeFrenchM-Contest-Big-Snowball-18238

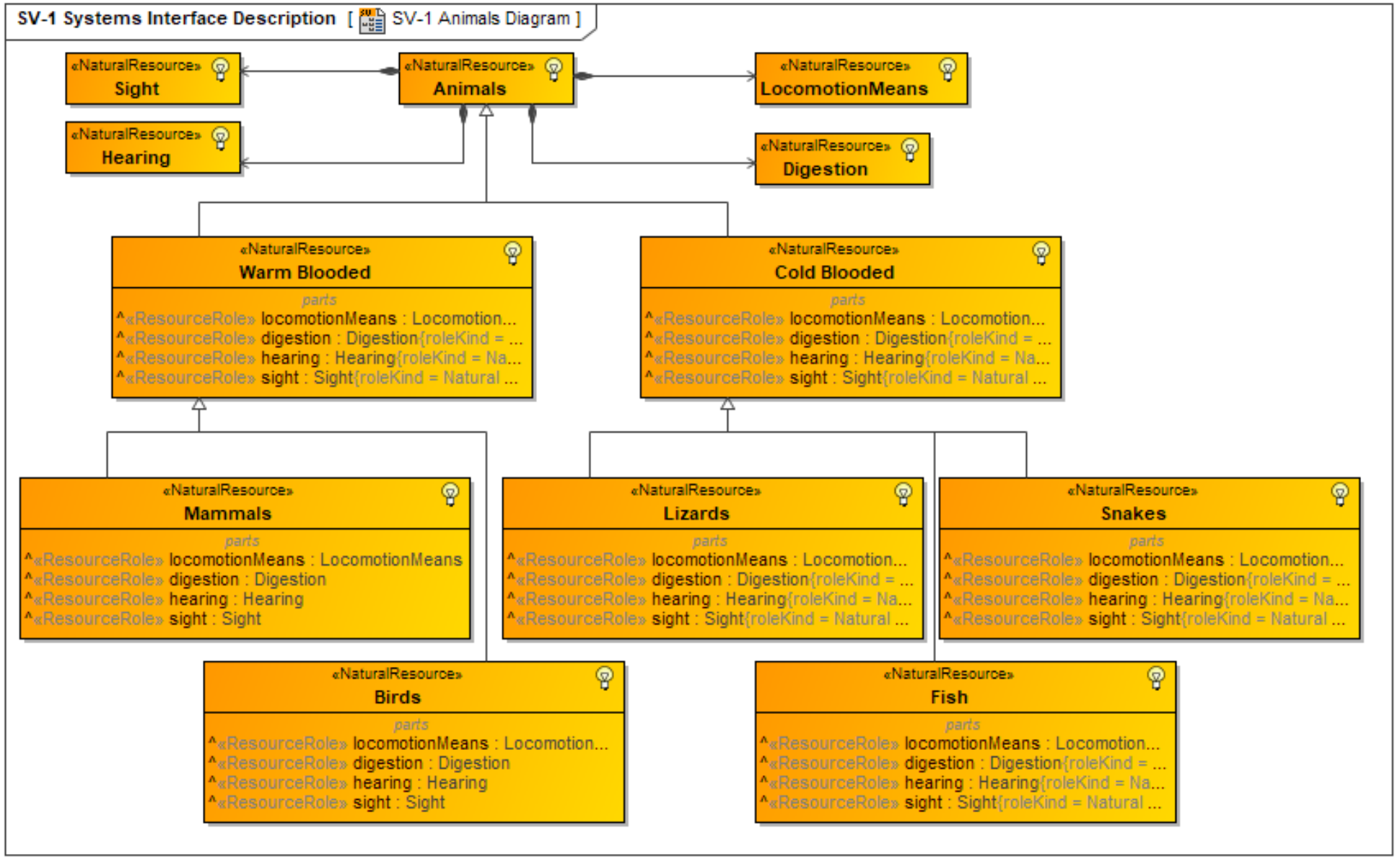
An Example Taxonomy in the Animal Kingdom

- Animals are warm blooded or cold blooded
- Warm blooded animals are either mammals or birds
- Cold blooded animals are lizards, snakes or fish
- The taxonomy implies that child classes have the same attributes as their parent class



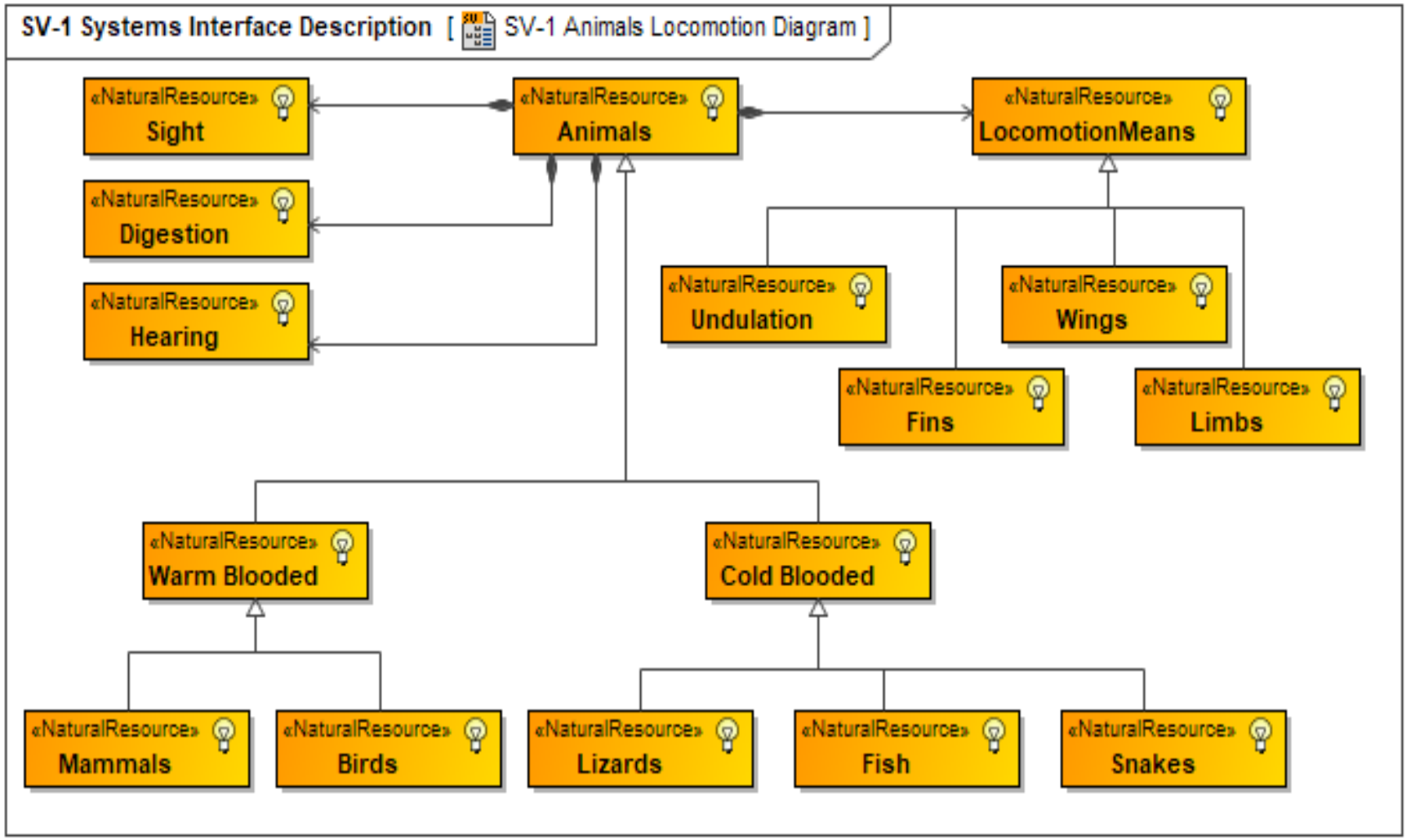
Inherited Properties

- Properties are inherited from the parent class
- Inherited properties both constrain and enable the child class



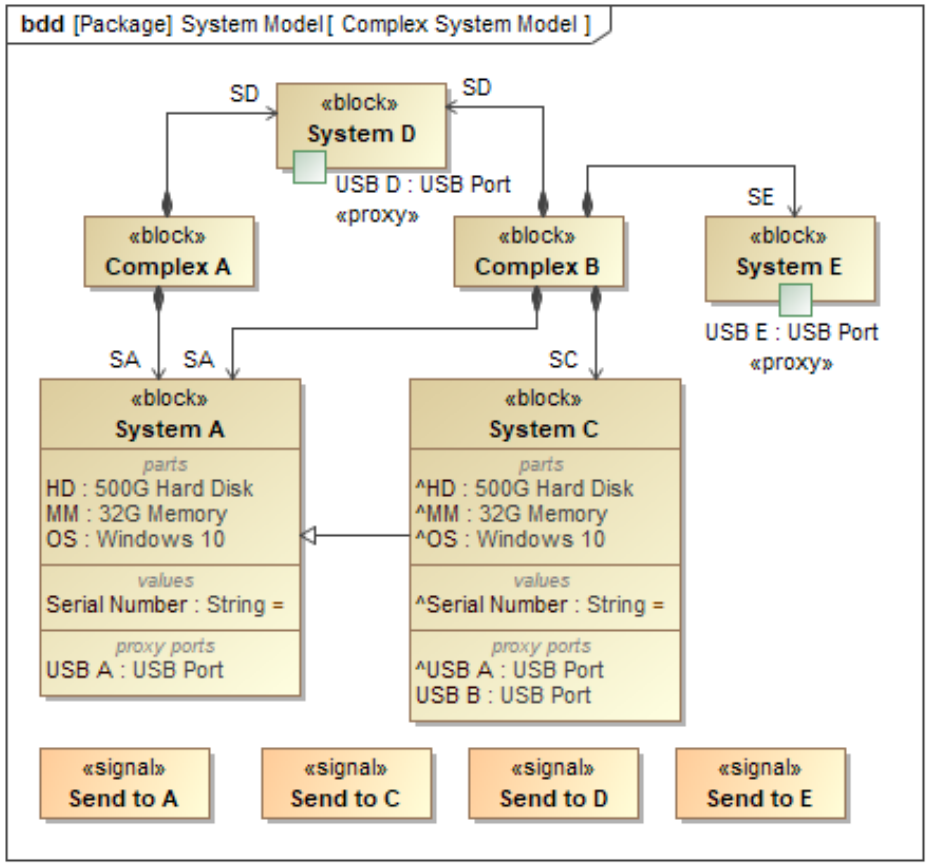
Problems With Inheritance and Where to put Attributes

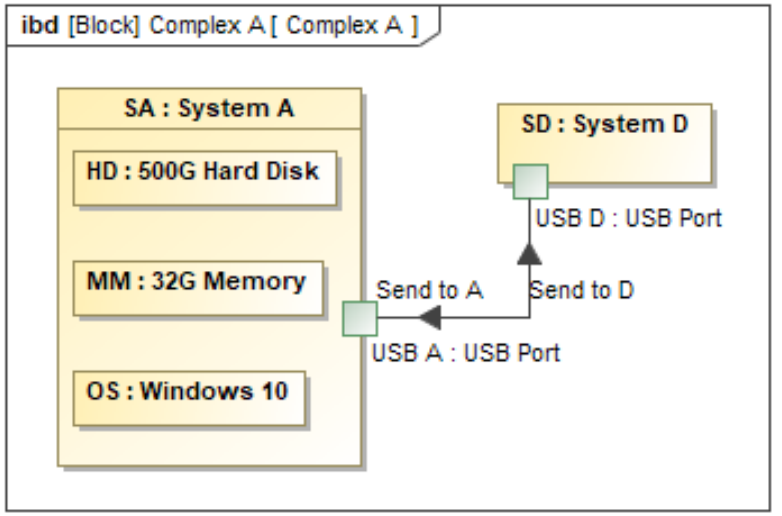
- Let's take the example of how to categorize locomotion.
- Locomotion has four types: wings, fins, limbs and undulation
  - Adding a property to a class means all child classes inherit that property.
  - So where do we attach them?
- All mammals have limbs, right?
  - Well, yes and no.
  - Marine mammals have fins.
  - Bats also have wings.
- All fish have fins, right?
  - Some fish such as the hagfish do not.
  - Mud fishes, Garnai, Axolotls, and loaches have fins and legs.
  - Flying fish can glide for up to 650 feet on their "wings".
- Duck-billed Platypuses, etc.

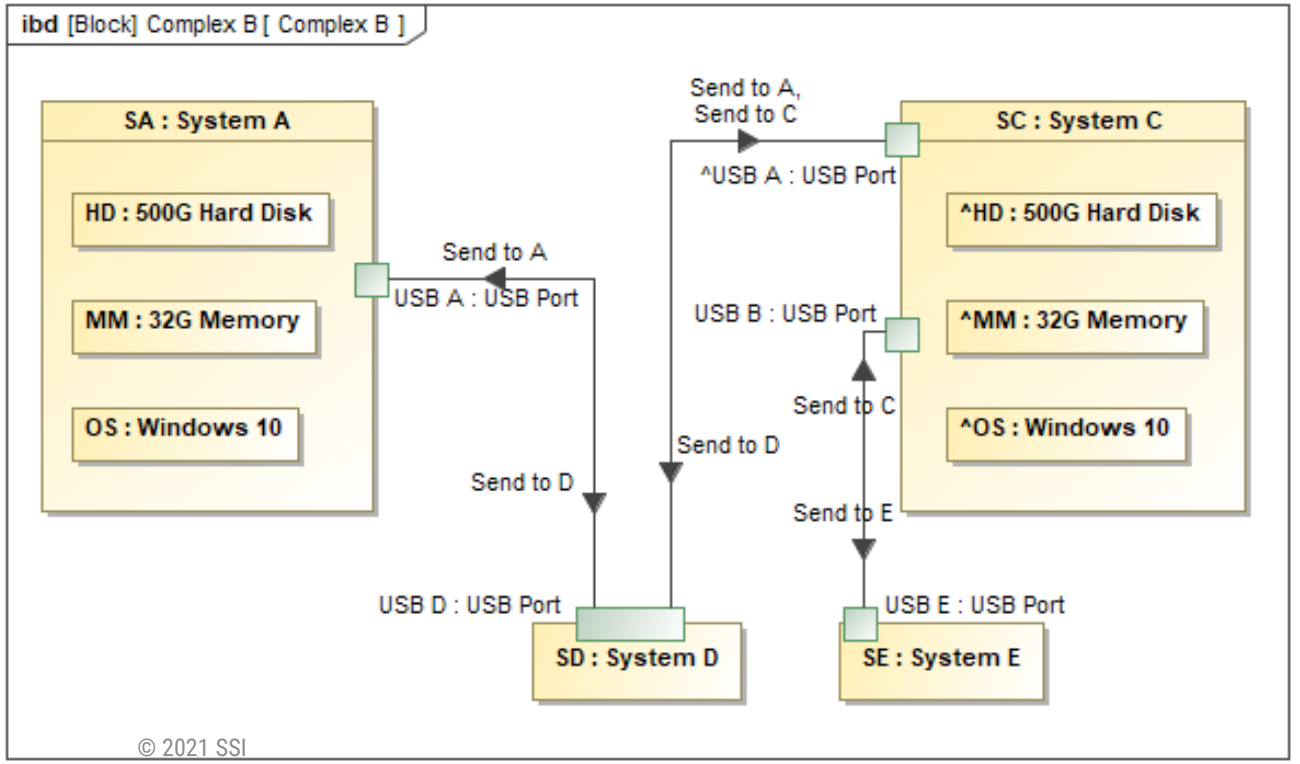


Use of Defined Systems in IBDs

- Complex A & B include System A & D. Complex B adds C & E
- Item flows defined between D & A in Complex A populated in Complex B
- Inherited USB A item flows to USB D can be populated, Send to C added
- New item flows remain within their contexts.

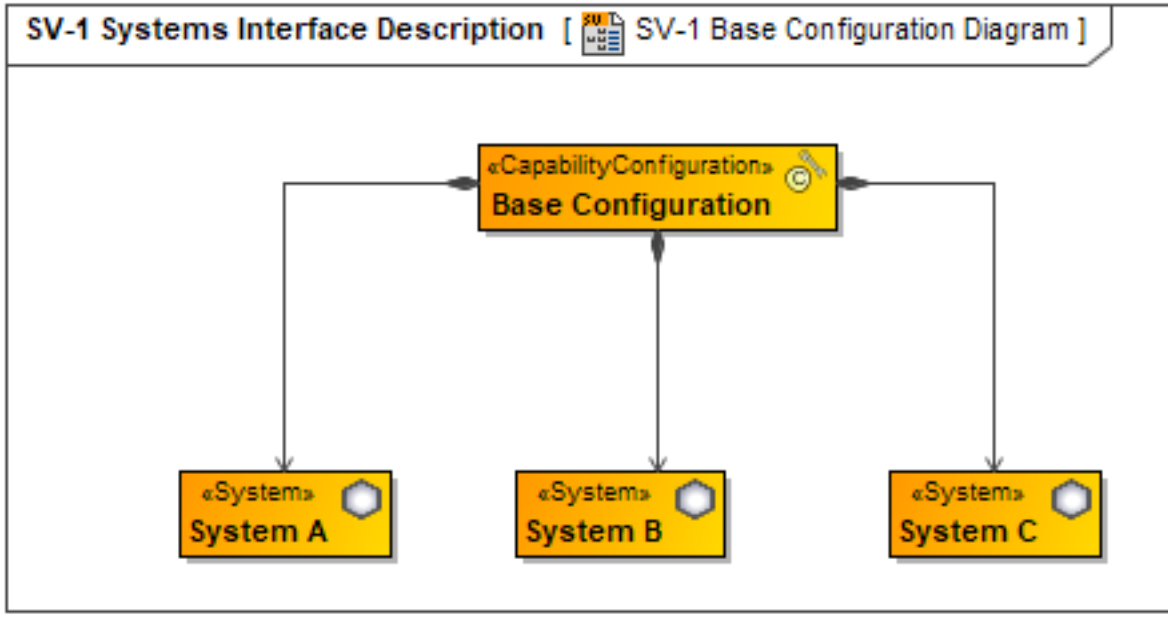


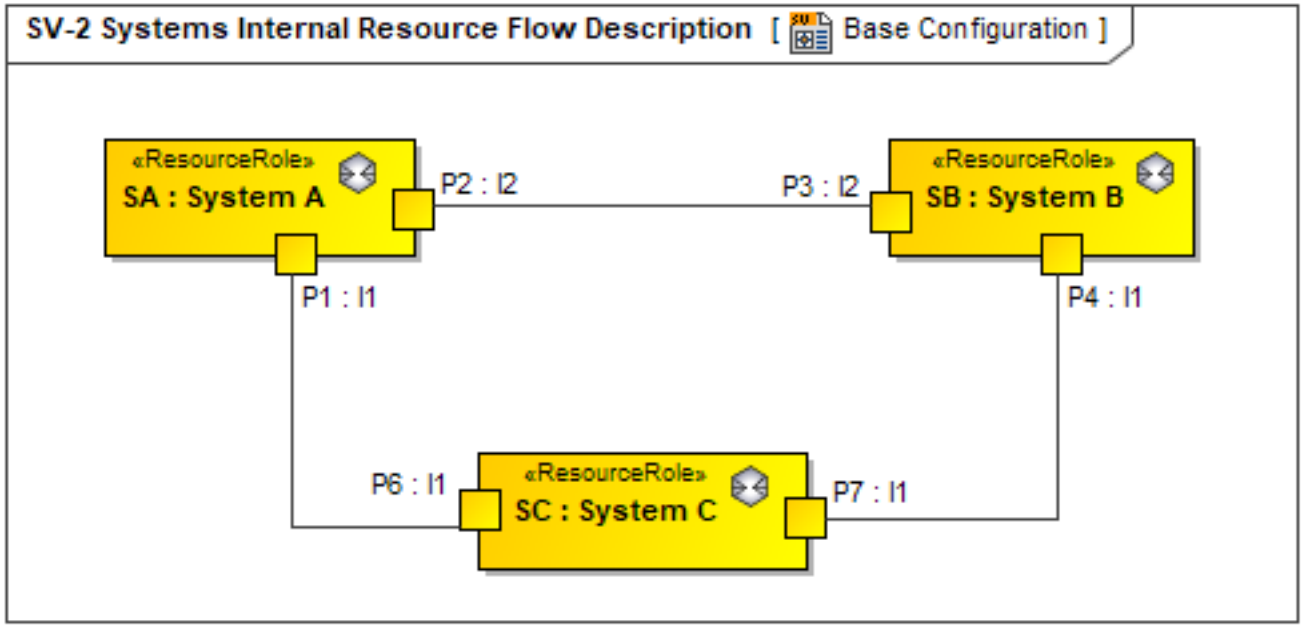




The Evolution of System Configurations

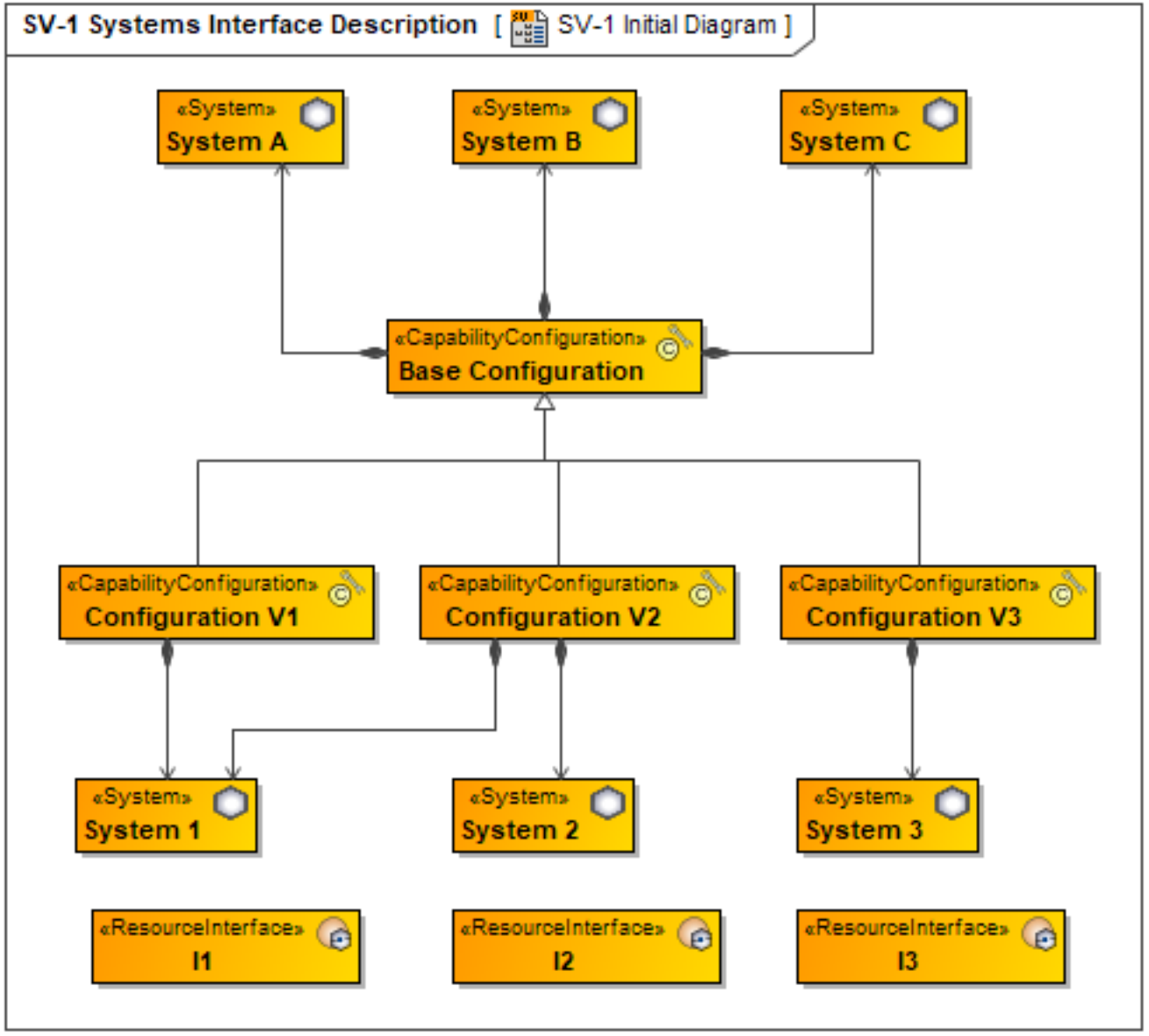
- UAF provides the ability to model systems and their evolution over time
- Adding new parts to a system configuration implies that it is a New system
- Example configuration:
  - The Base Configuration is made up of Systems, A, B & C
  - The internal resource flow diagram configures them as shown below
  - The different systems each contain two resource ports and connectors.





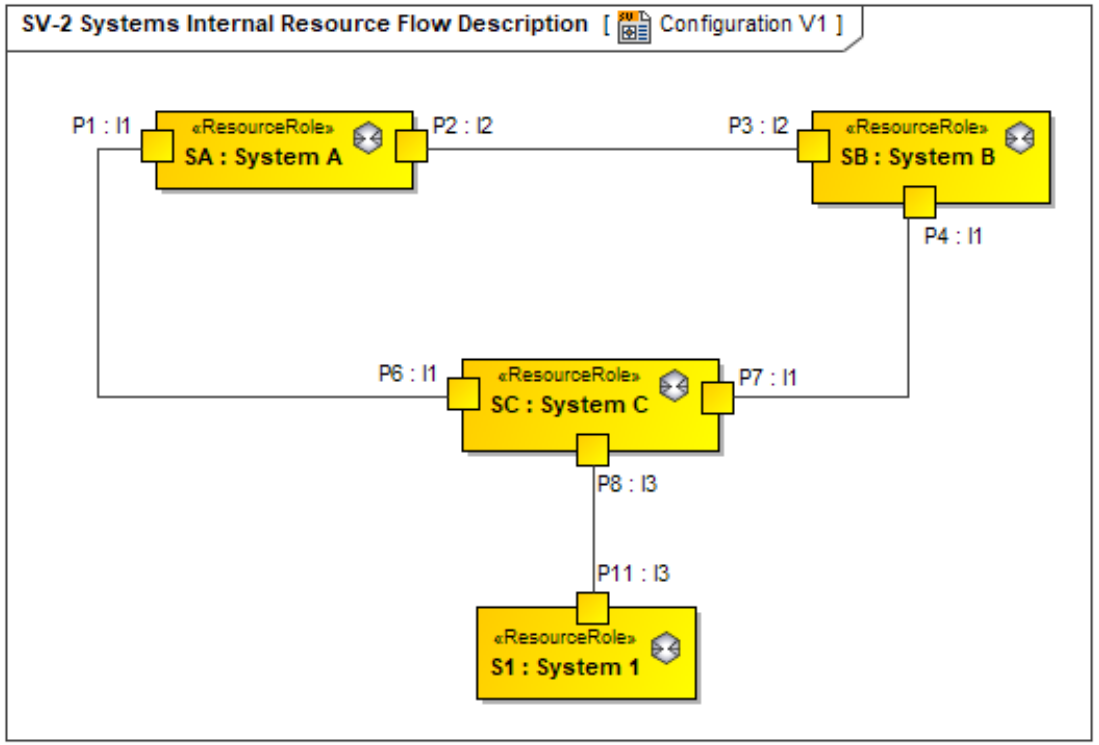
Configuration Modifications

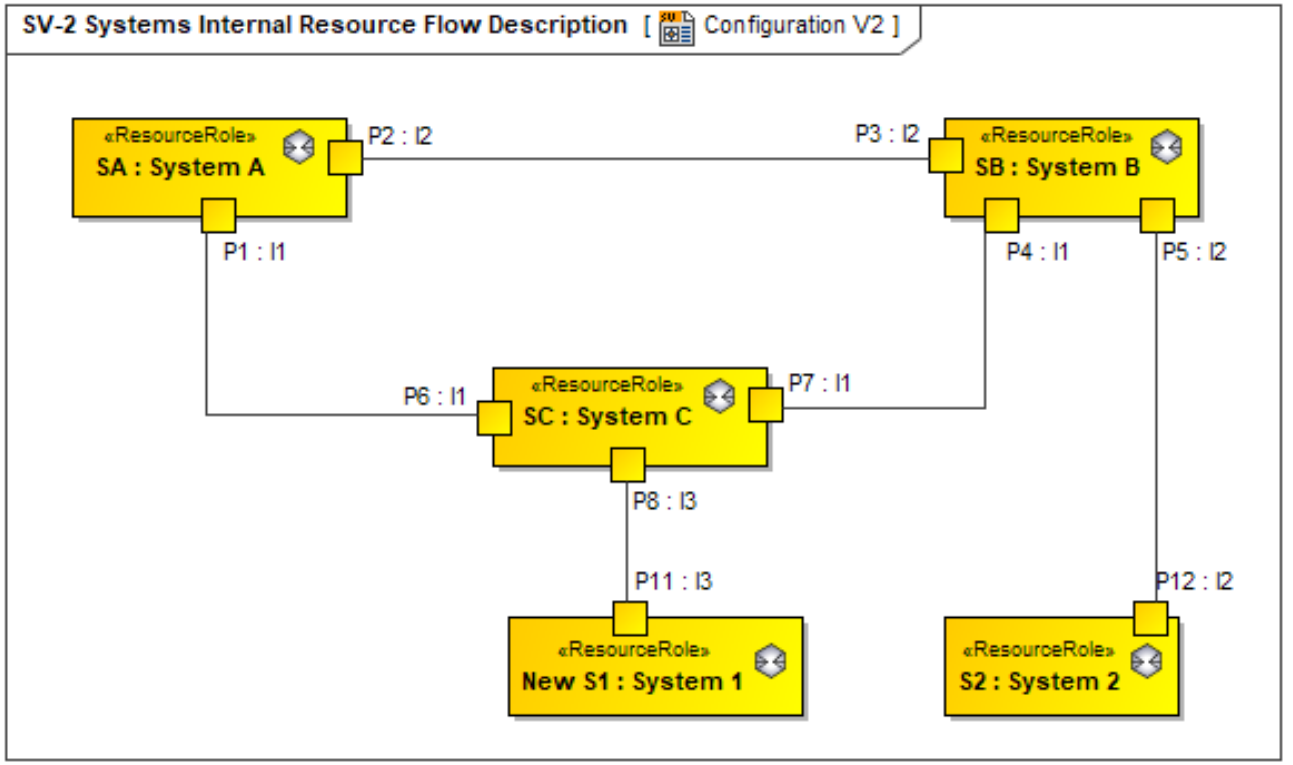
- The Base Configuration will be modified over 3 phases resulting in the following:
  - V1 adds System 1
  - V2 adds Systems 1 & 2
  - V3 adds System 3
  - Systems A, B & C are in all 3 configurations, so we have V1-V3 child classes of the Base Configuration.
  - This relationship as well as the connections are inherited as shown in the following diagrams.



Internal Configurations

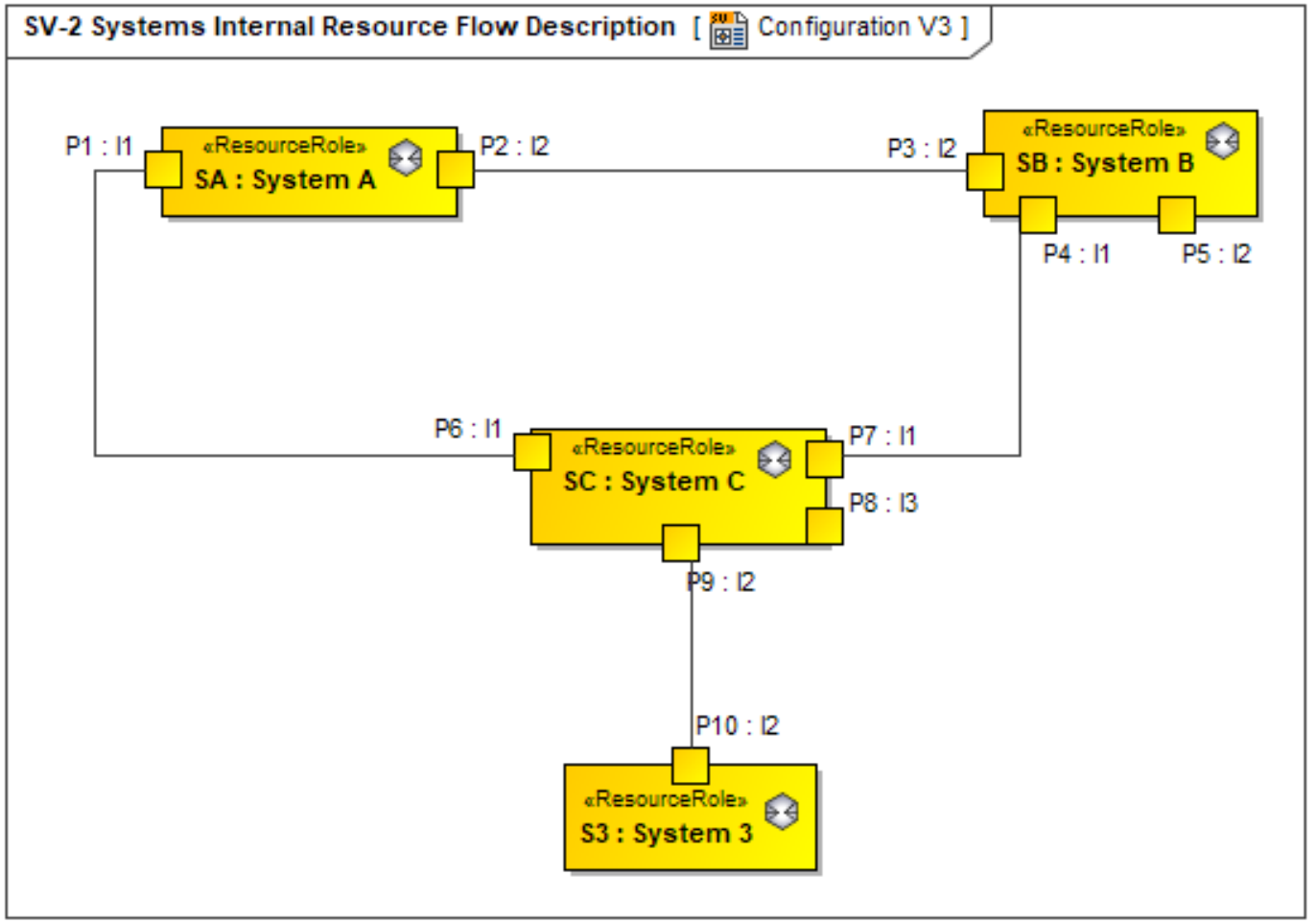
- The new configurations have modified Systems B & C
- System is 1 added to the configuration V1, Port P8 is added to System C
- Systems 1 and 2 are added to the configuration V2, Port P5 is added to System B





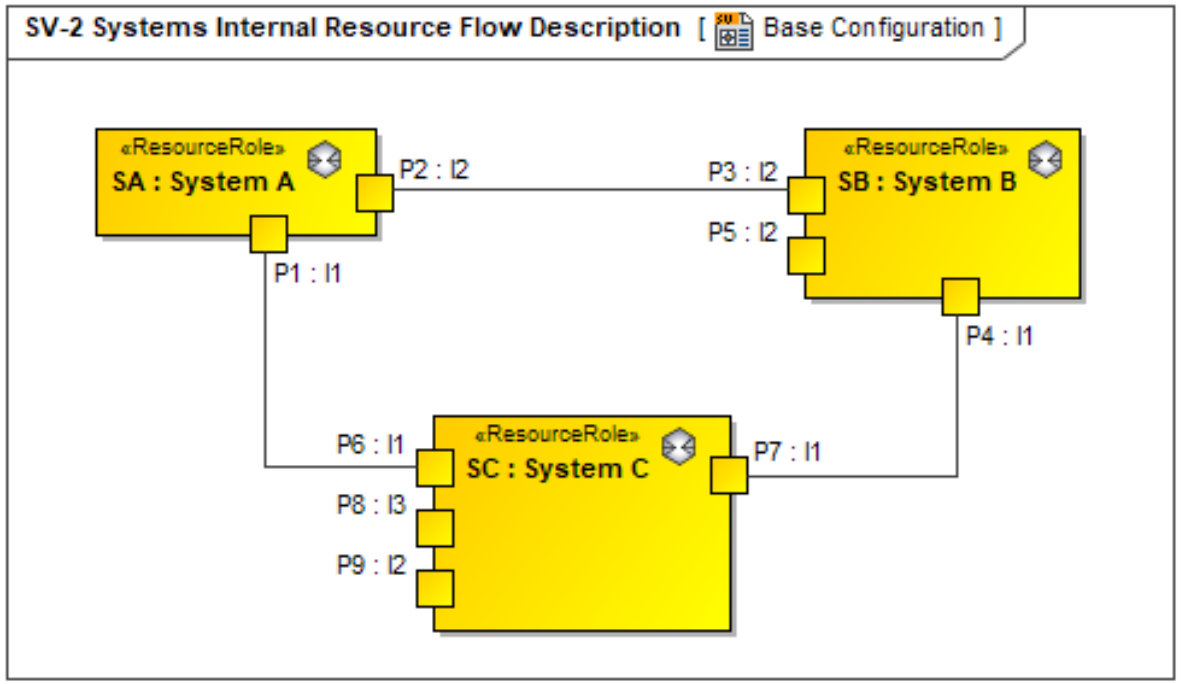
The Problem with Additive Components

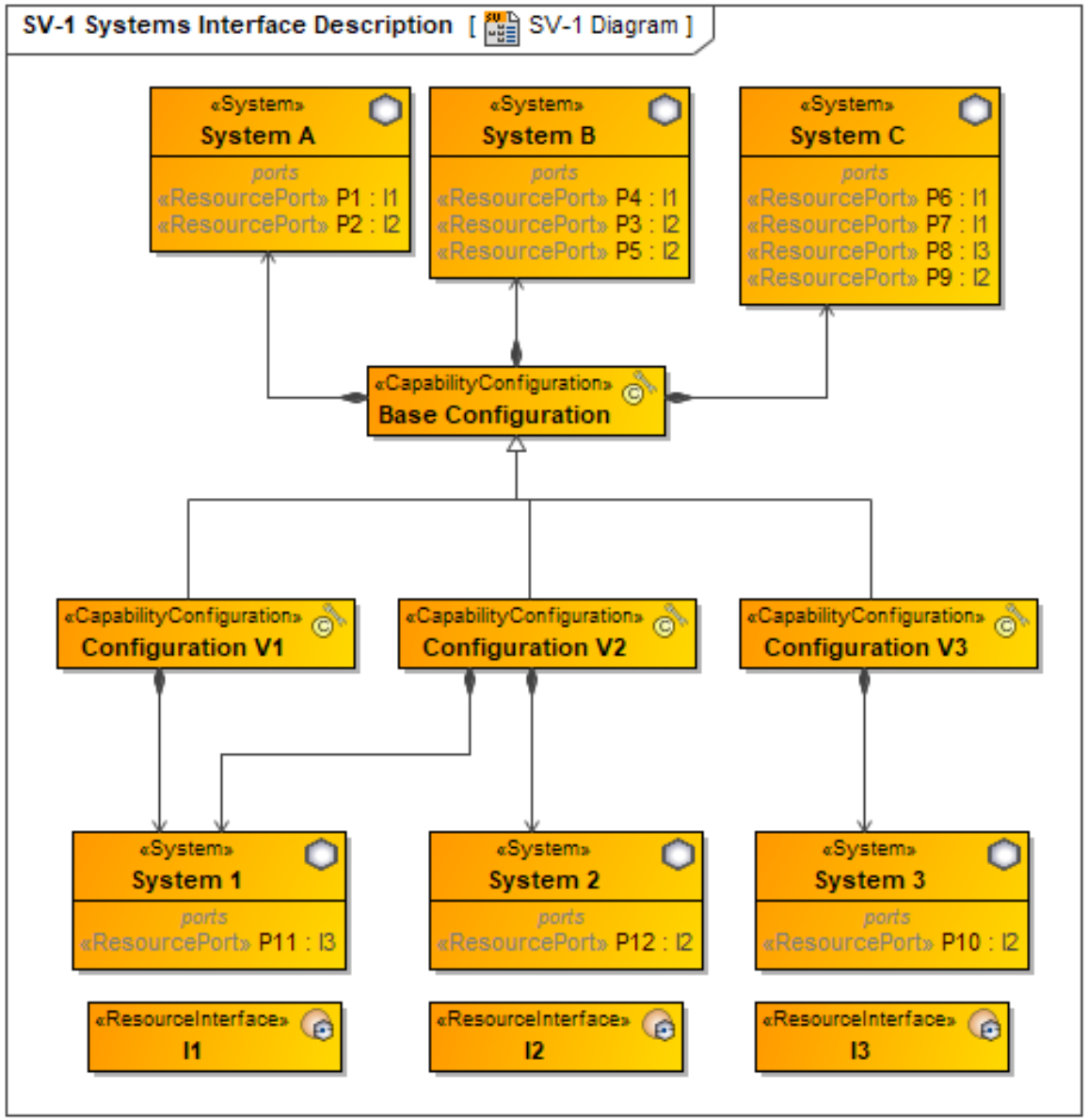
- System 3 has been added, Systems 1& 2 have been removed.
- Ports P5 & P8 added in configurations V1 and V2 are still parts of Systems B & C
- Not a big deal if just an extra port?
- What if they were:
  - Large/heavy systems?
  - Safety critical systems?
  - Energy hungry systems?
  - Unwanted software?
  - Creating an access vulnerability?
  - Etc.



Resulting System Configurations

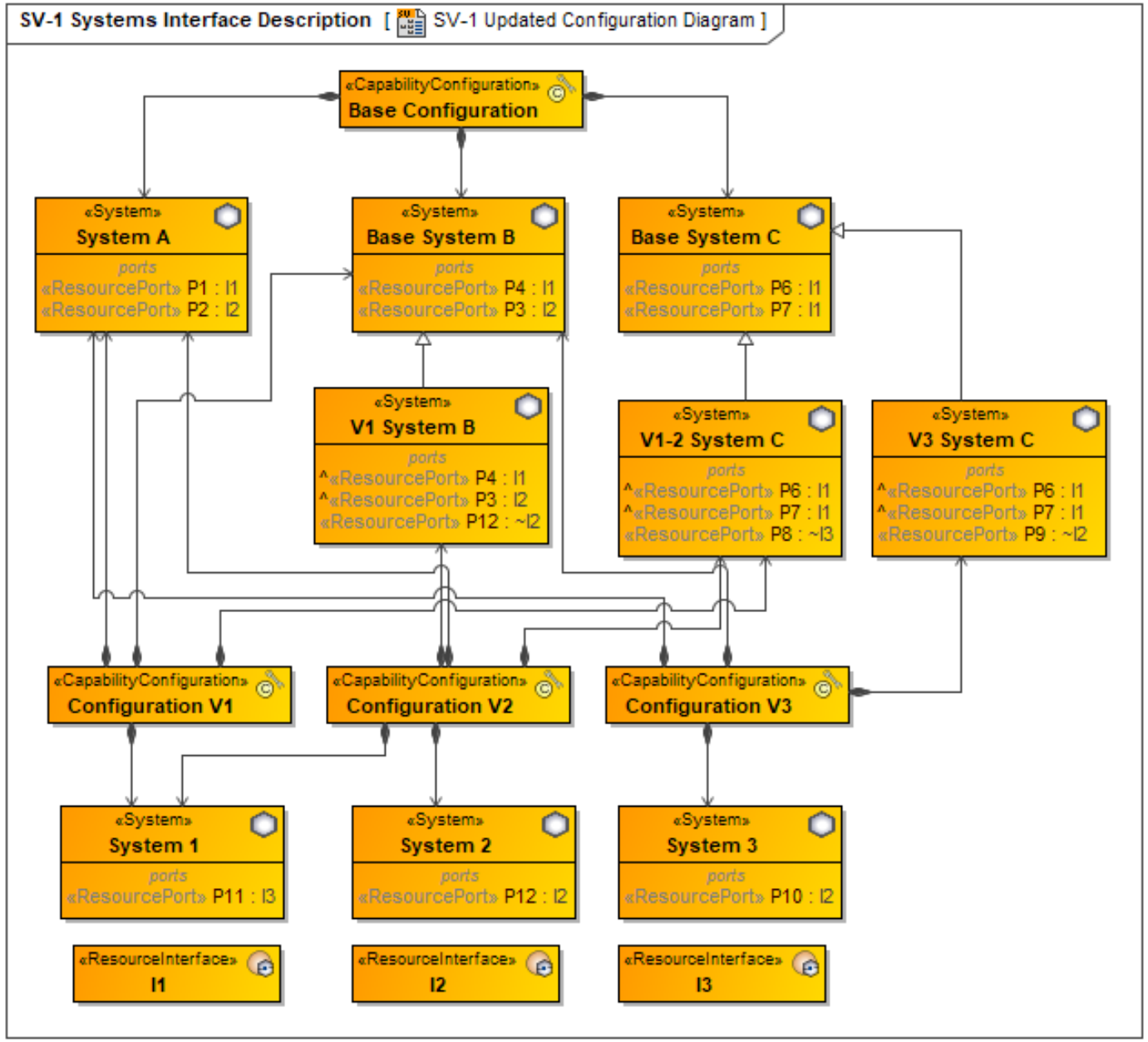
- The figure on the right shows the different components and their parts and ports.
- The figure below shows the repopulated Base Configuration diagram.



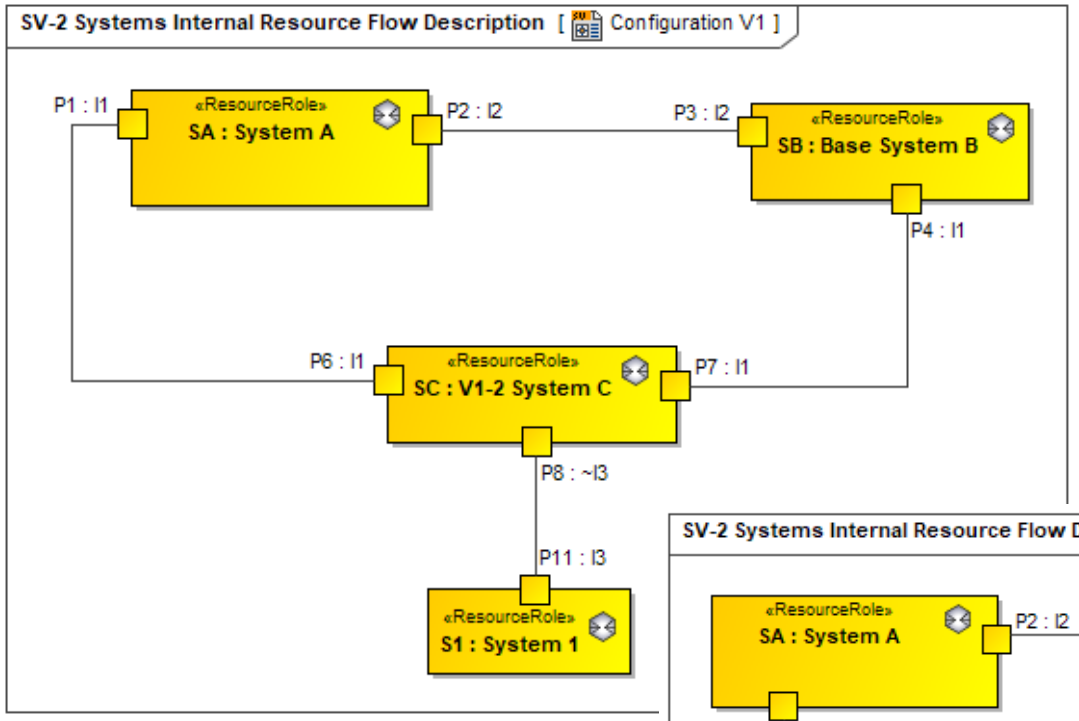


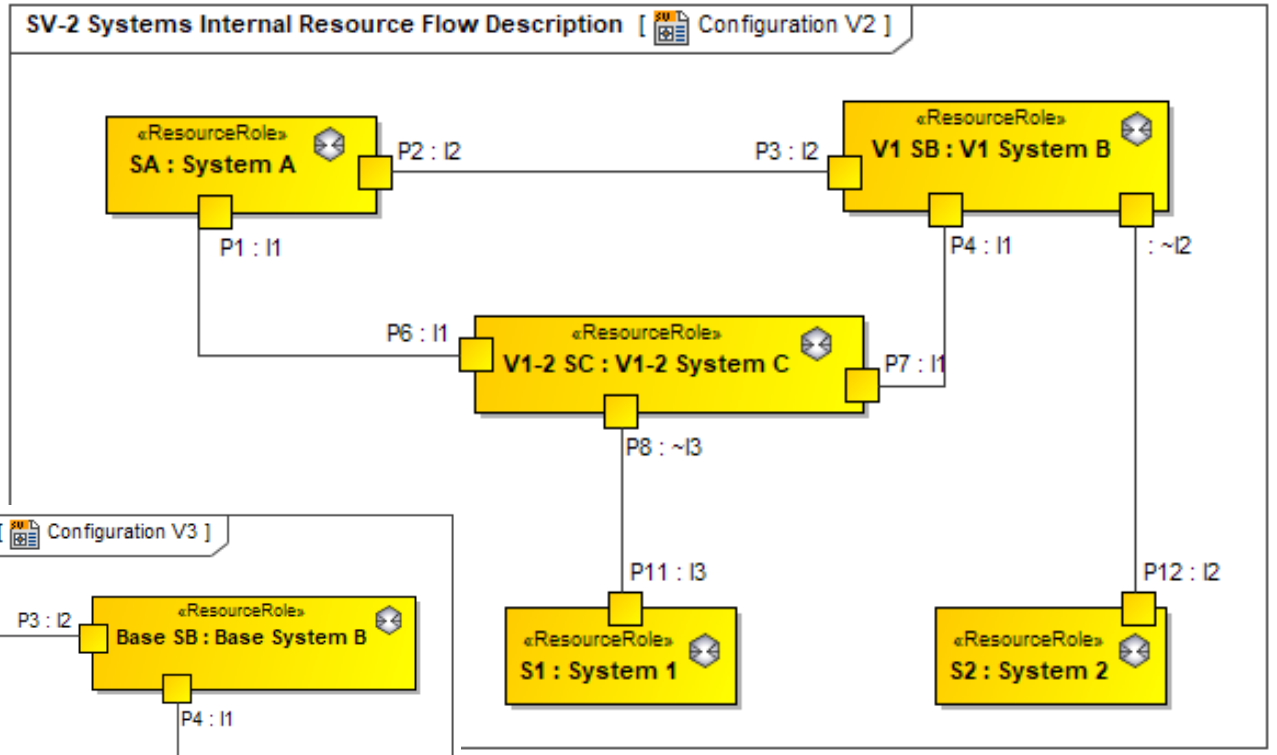
Updated Configuration

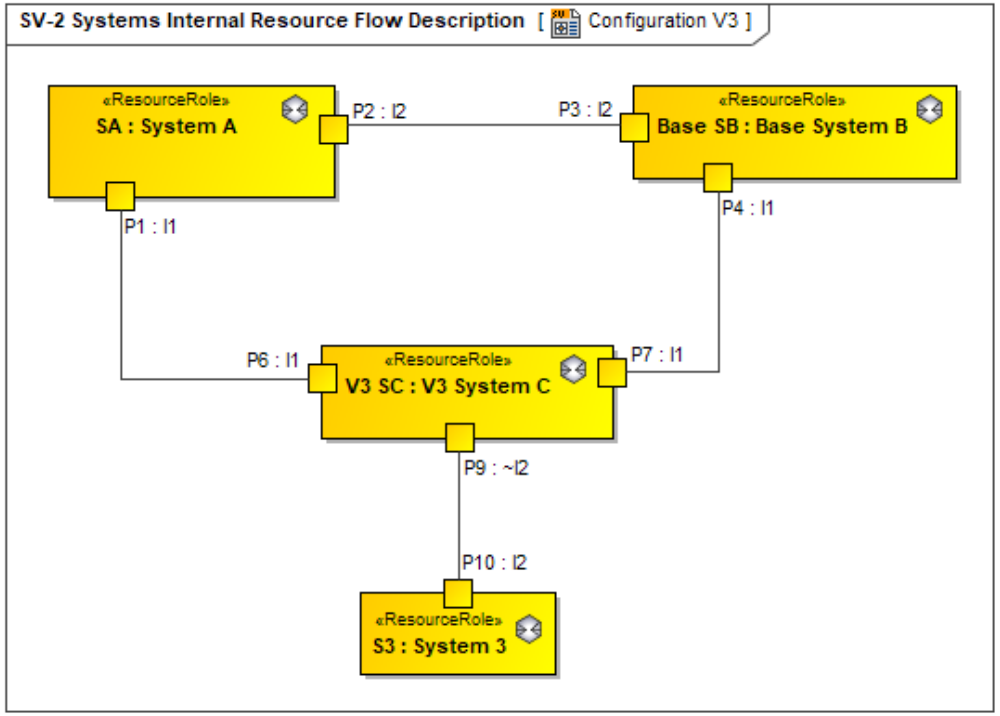
- Inheritance from the base configuration has been removed.
  - System A is the only common system between the 4 configurations, so, inheritance is of no benefit and in fact complicates things.
  - However, systems B & C can benefit
  - Common elements are in the parent class.
  - Child classes add new ports.
- The Base configuration contains the “original” versions of Systems A, B and C
  - Configuration V1 contains System A, Base B, and V1-2 system C
  - Configuration V2 contains System A, V1 System B, and V1-2 system C
  - Configuration V3 contains System A, Base B, and V3 System C



Internal Configurations







- This modified structure avoids unwanted parts, maximizes reuse and creates fit for purpose structures.

Example Model

DoDAF/UAF and System Changes over Time

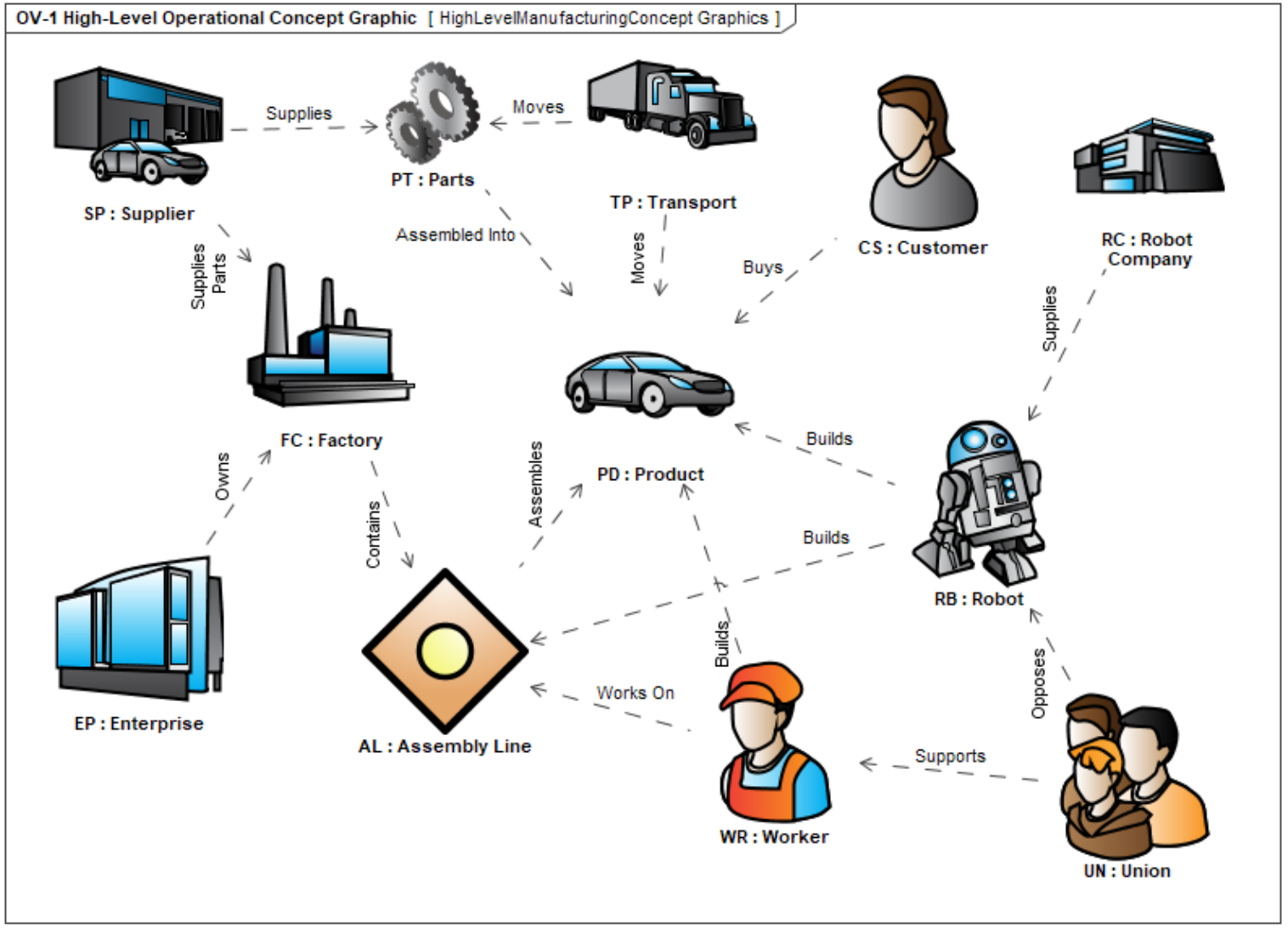
- The UAF Resource interaction diagrams are used to define system structure
  - Shows how resources (systems, roles, posts and organizations) interact
  - Created from systems, system nodes and organizations
  - Defined system configurations can be linked to project deployments
  - System configurations will change over time
  - Each configuration is modelled in a separate context
  - ***However, changes to a system in context affects the system in all contexts***
- Project views link configurations to time and capability.
  - Actual elements are also linked.

Example Automotive Factory Model

- Problem Statement: Powerhouse Engines (PE Inc.) is an automotive supply company providing internal combustion engines. PE Inc. finds that it has gradually become less competitive over the years largely due to their outdated technology and largely manual processes. Foreign and domestic competitors have started to cut into their business and the stakeholders are concerned that the company's loss of market share will accelerate and that they will eventually become insolvent. To combat this, the shareholders have proposed an investigation into strategies and technologies such as Augmented reality, Robotic assembly systems, 5G, AI, Additive manufacturing, outsourcing of select manufacturing and IT systems, Battery technology, Data analytics, Hybrid/electric engines, etc. These technologies will be rolled out over a 3-phase technology deployment plan.

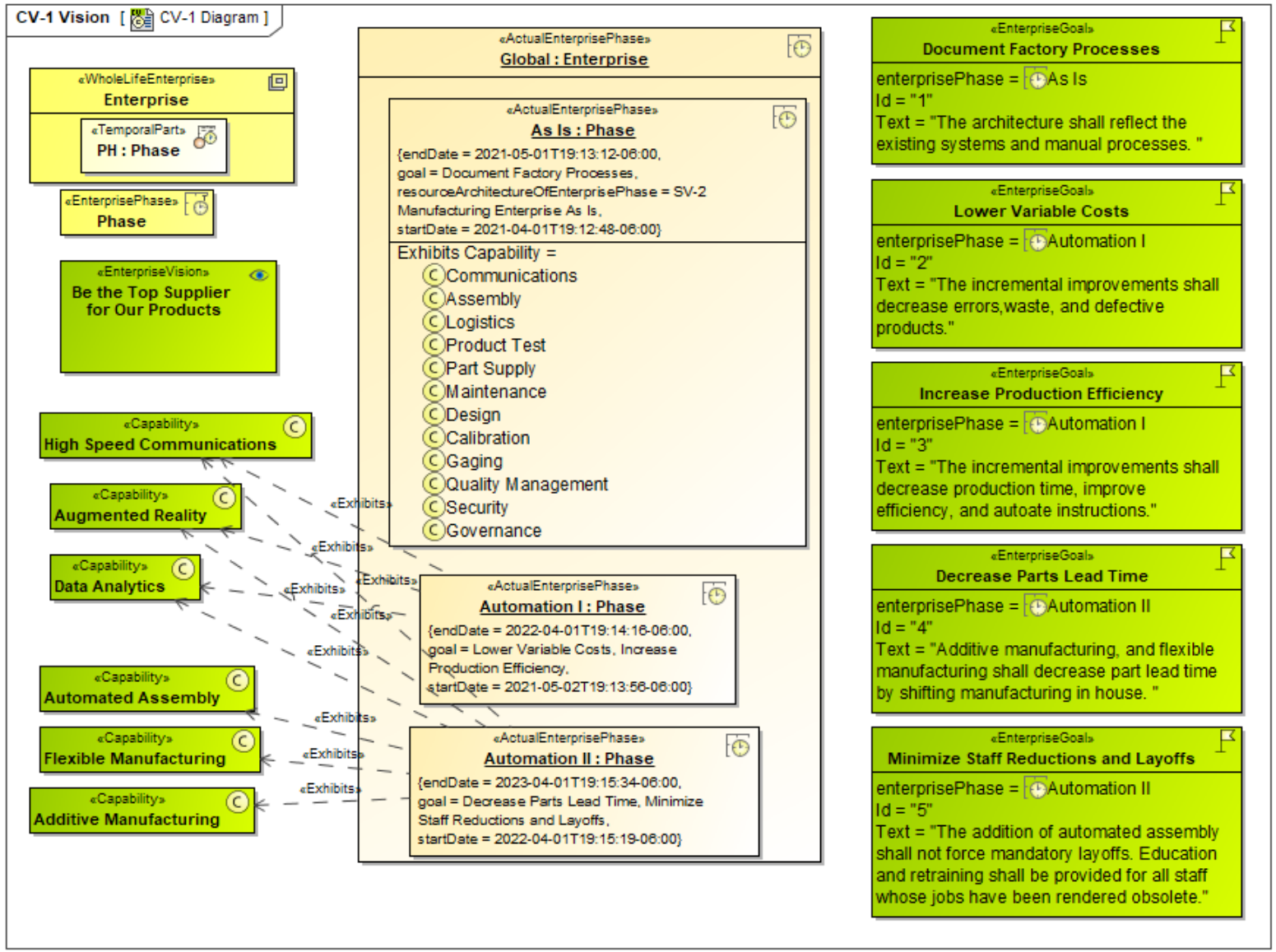
High Level Manufacturing Concept for Powerhouse Engines

- Defines the high-level concepts for Powerhouse Engines
- The part supplier could be an external company, an internal casting department, or an in-house 3D printer.
- All 3 will be deployed over the 3 phases of technology introduction.
- Robots and other technologies will be introduced as well.



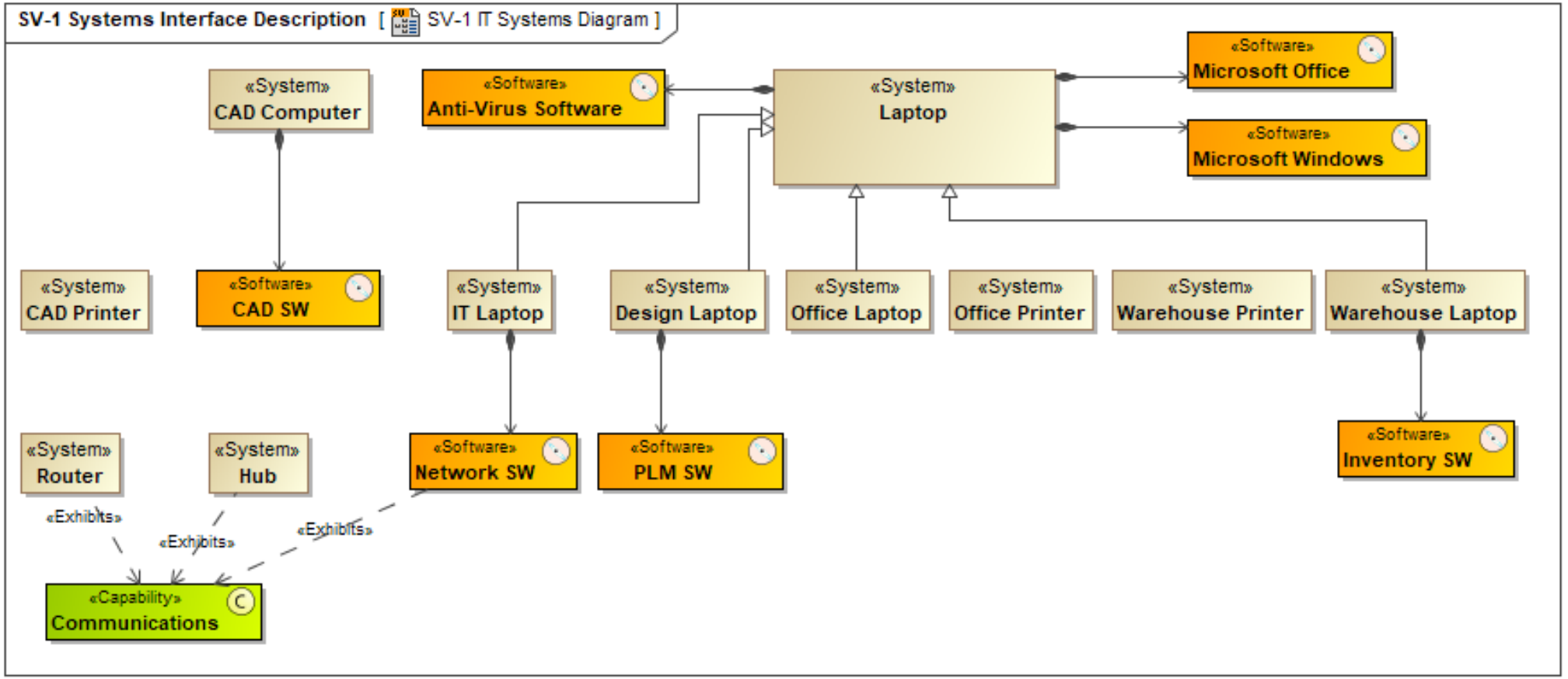
Whole Life Enterprise Phases

- The Global Enterprise is divided into 3 phases: As-Is and Automation I & II
- Existing capabilities are documented in the As-Is phase
- High-Speed Comms, Augmented Reality, and Data Analytics in Auto I
- Automated Assembly, Flexible Manufacturing and Additive Manufacturing in Auto II
- Additional Goals for these phases are also assigned to the phases
- Systems will be defined to implement the Capabilities and Goals



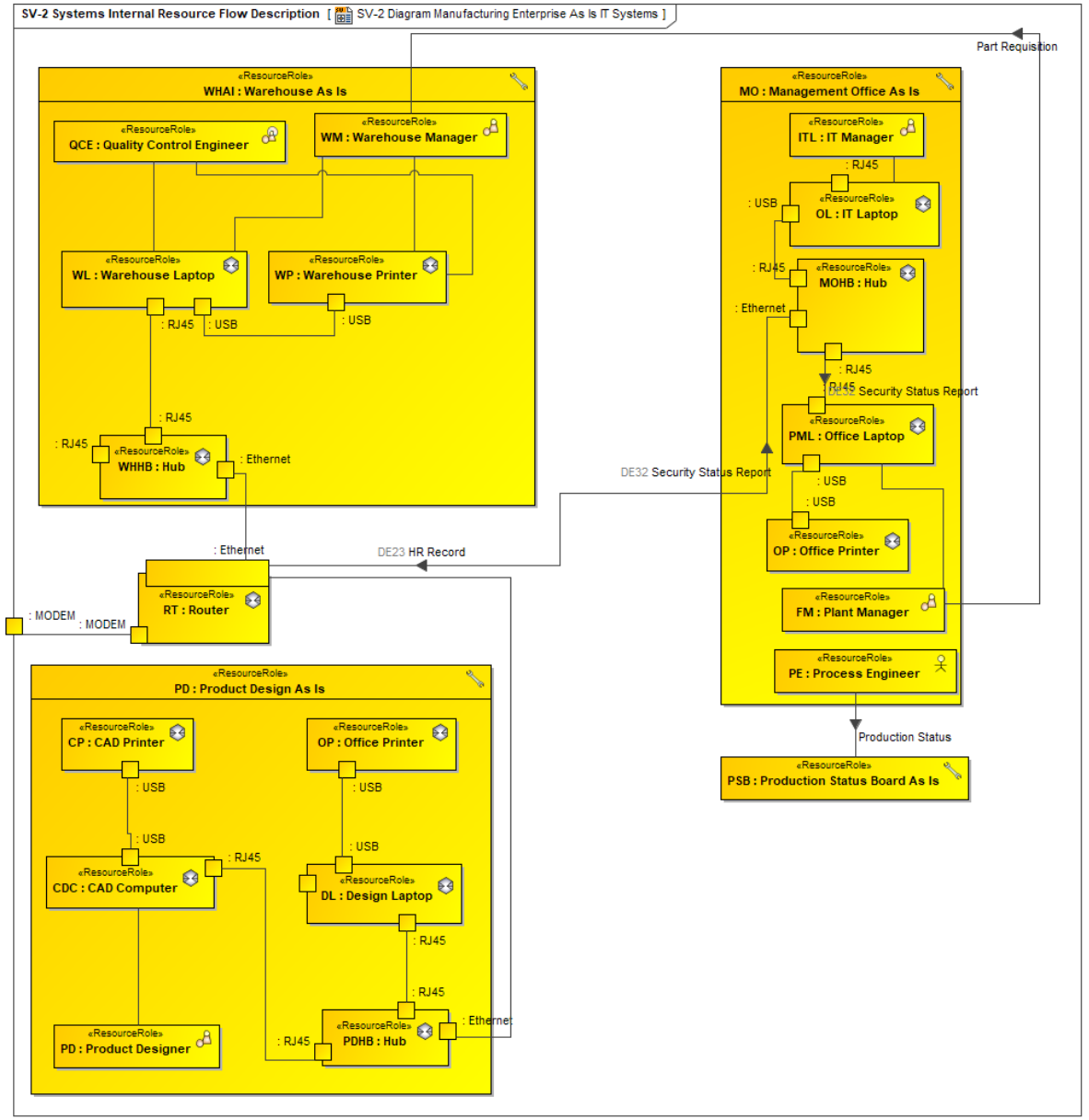
Powerhouse Engines IT Systems for the As-Is Phase

- The existing IT Systems are fairly primitive containing USB and ethernet ports
- Common software is defined for all laptops
- Specialized hardware and software are defined for other systems.



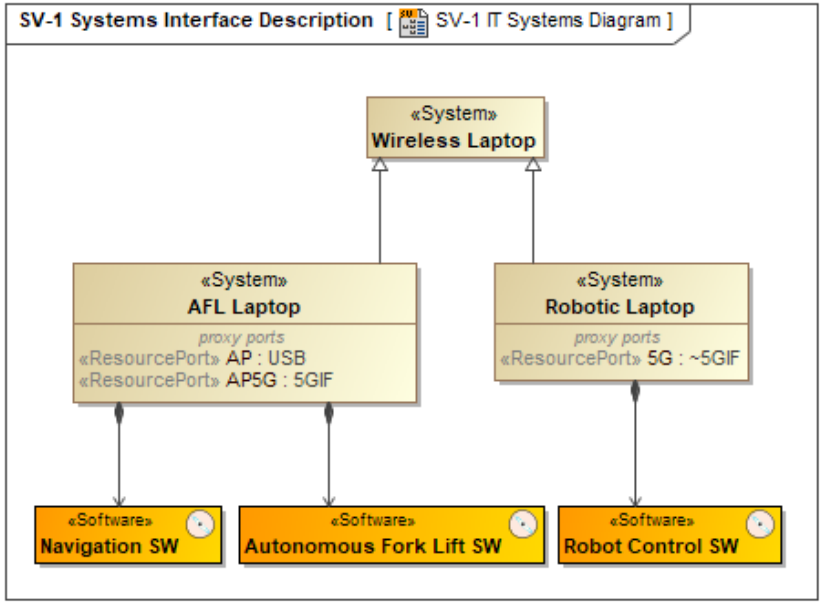
IT Systems Integrated into the As-Is Factory Configuration

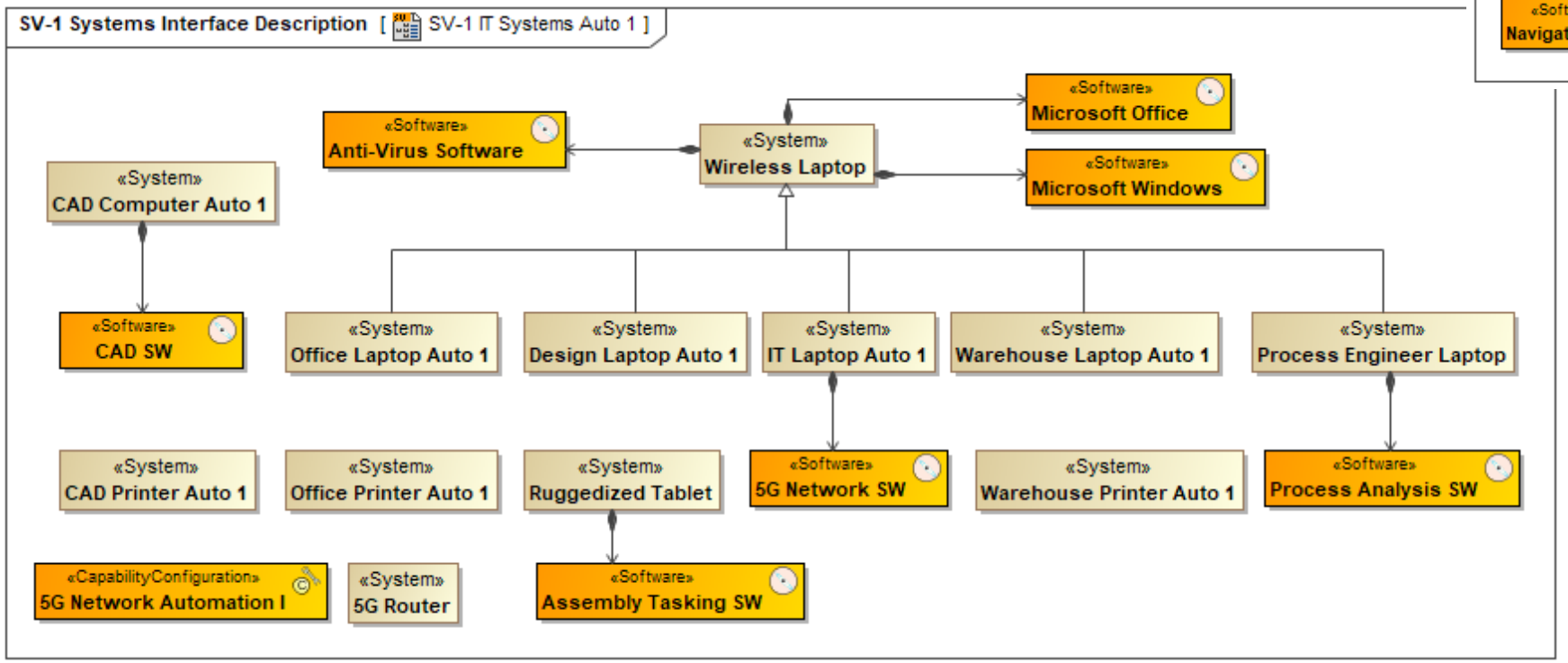
- Three areas have been defined for the factory:
  - Warehouse
    - Inventory management, procurement, parts management and distribution
  - Product Design
    - CAD designs for systems and requirements management
  - Management Office
    - Factory management, IT management and general management
- Existing systems and interactions are defined



IT Systems for Automation I & II

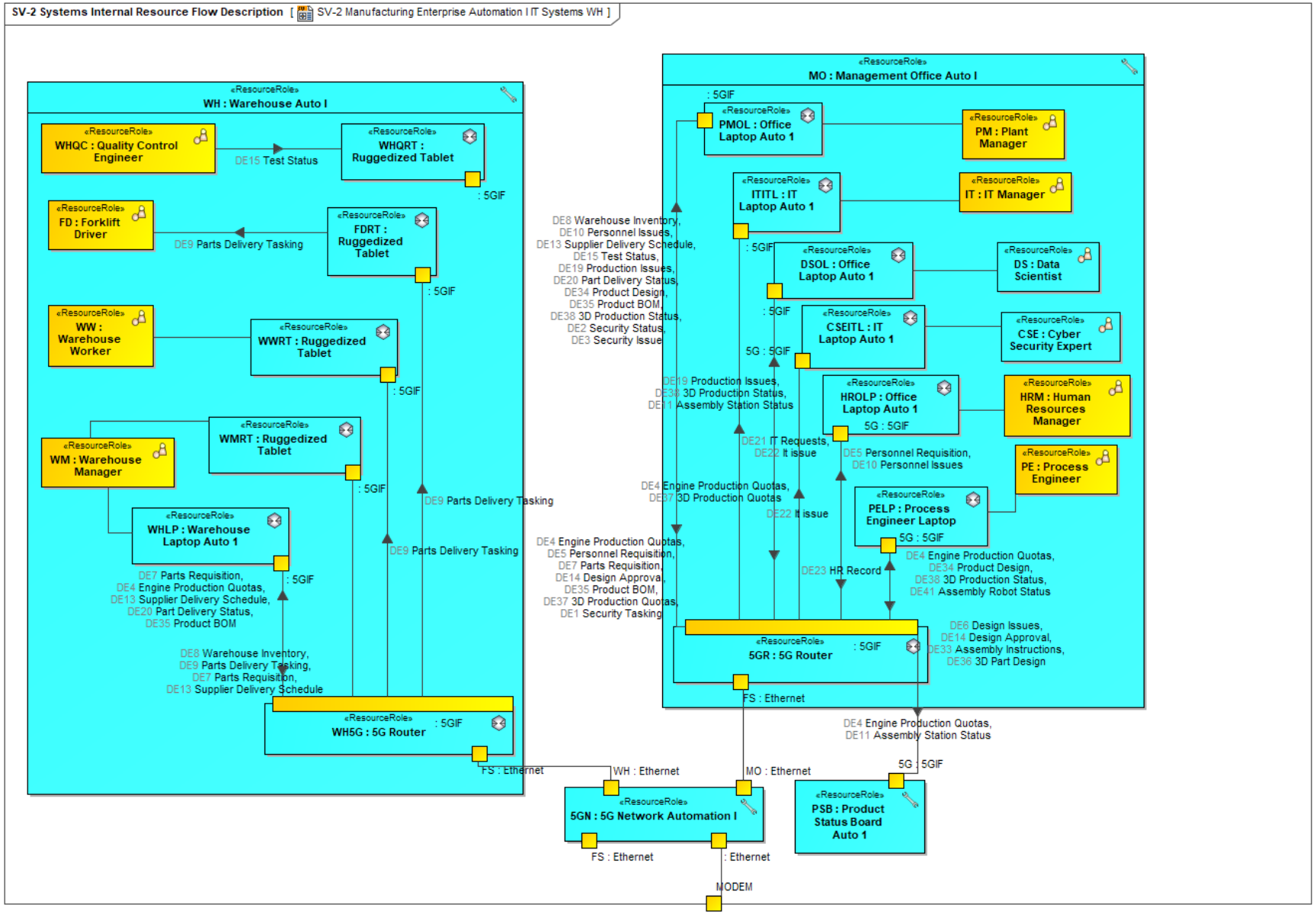
- The Laptop has been redefined as a Wireless Laptop as the factory is going wireless
  - Specialized laptops for the major functional areas have been defined and ruggedized tablets have been distributed
  - The 5G network has been deployed in the factory
  - The Process Engineer has a specialized laptop
- Dedicated laptops were added in Auto II for the Autonomous Forklift & Robot





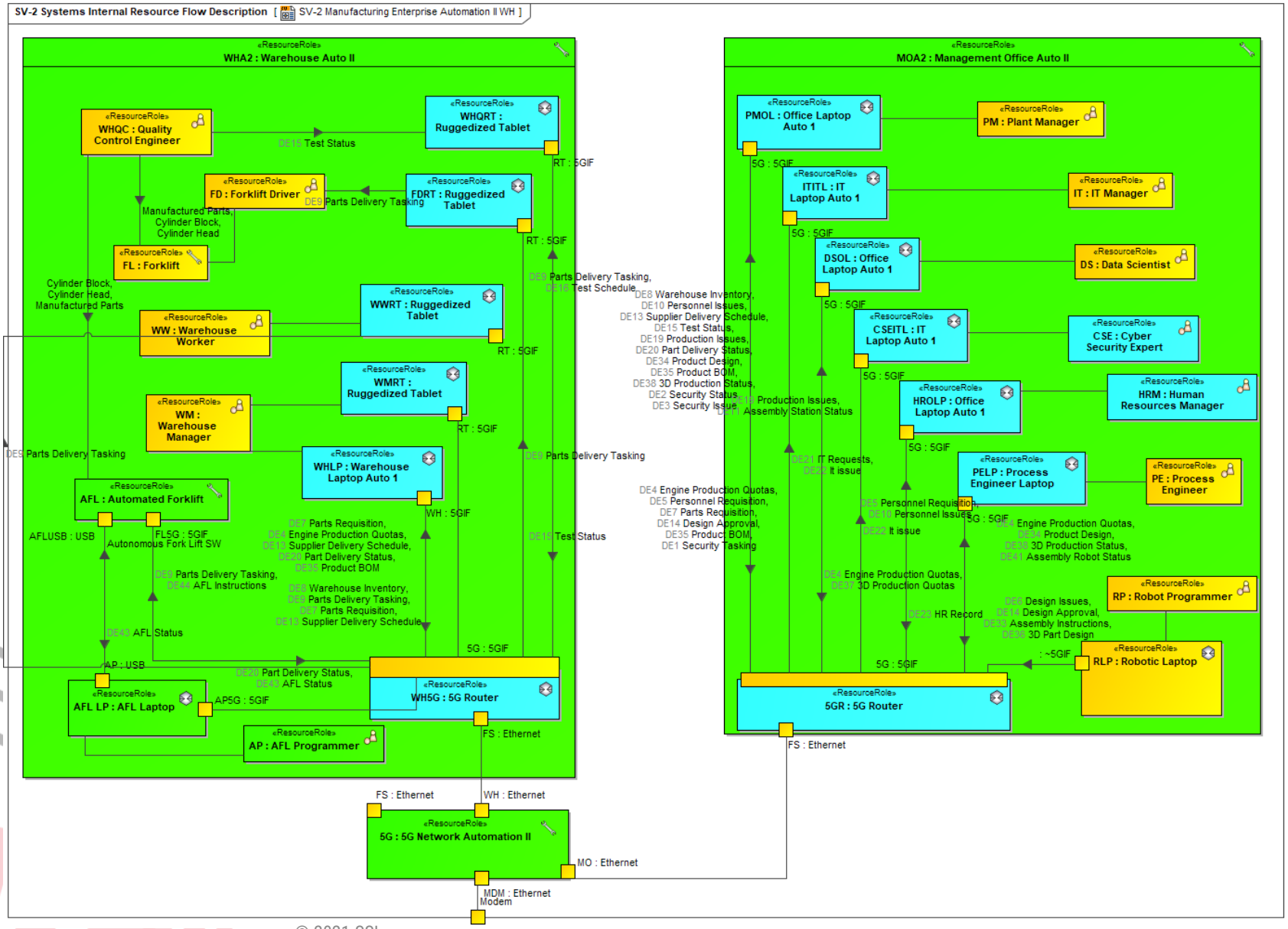
Factory Configuration for Automation I

- The diagram has been color coded to differentiate elements from the different phases
  - Yellow for As-Is and Blue for Auto I
- Everything except the people is new.
- The owning systems: Warehouse and Management Office are also new
  - As most of their components and interactions have changed, new systems need to be defined.



Factory Configuration for Automation II

- Elements added for Auto II have been colored green
- New people and systems have been added.
  - Automated Forklift (AFL), AFL Programmer, and AFL Laptop to program and control the AFL
  - The Robot Programmer and Laptop have been added to control the Robot Assembler (Not Shown)
- The owning systems: Warehouse and Management Office are also new
  - Many of their components and interactions have changed, so new systems need to be defined.

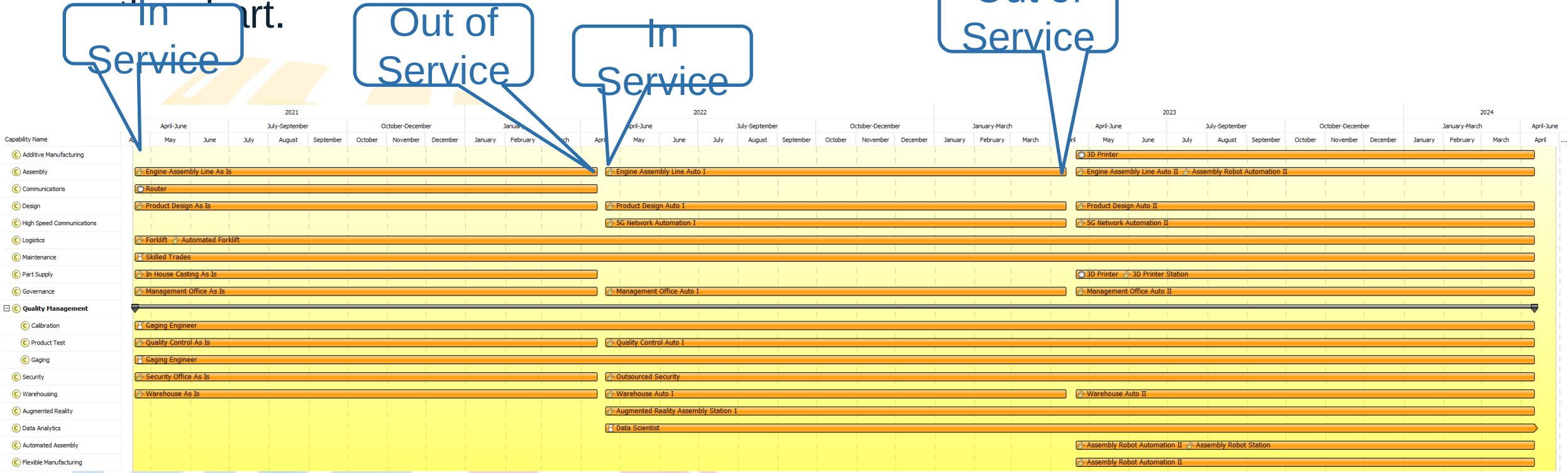


CV-3 Capability Phasing [St-Rm]

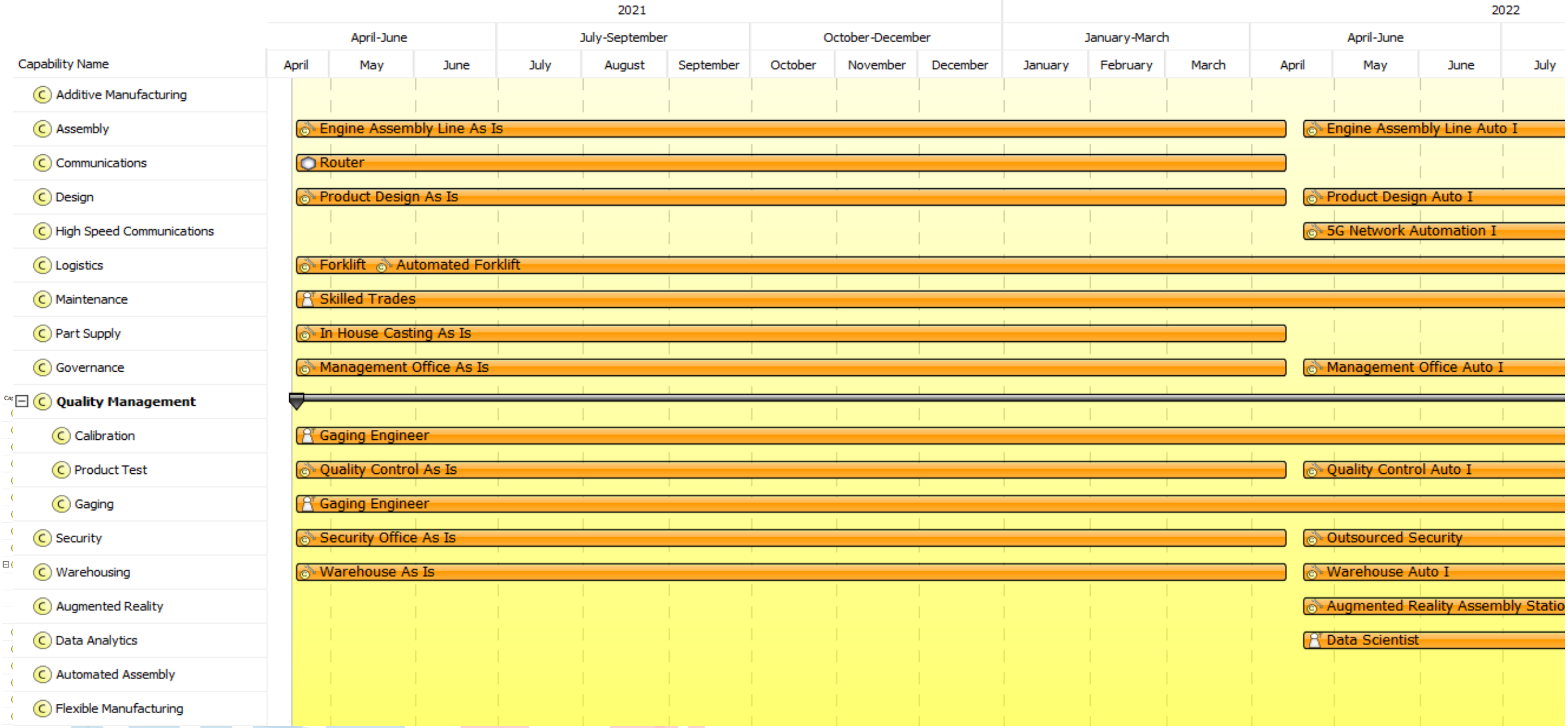
- Stakeholders: PMs, Executives, Enterprise Architects.
- Concerns: capability deployment to organizations over time.
- Definition: addresses the deployment of capability(ies) to actual organizations over time.
- The CV-3 (St-Rm) addresses the planned achievement of capability at different points in time or during specific periods of time, i.e. capability phasing.
  - A generated report showing the capabilities, the systems that realize these capabilities and when they will be deployed and taken out of service.
  - Information for this report is defined using the Actual Projects diagram, measurements diagram, and the CV-2 Capability Taxonomy diagram.

CV-3 Capability Phasing [St-Rm]

- Start and end of Gannt chart lines are in service milestones and out of service milestone types.
- Linked to actual resources and resource types linked to capabilities.
- Automatically generated from information in over views.
- Milestone start and end times can be modified by dragging the Gannt ends on the chart.



CV-3 Capability Phasing [St-Rm]



33rd Annual INCOSE international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023

www.incose.org/symp2023
#INCOSEIS