

Continuous MBSE Modeling

Armstrong Process Group, Inc.
www.aprocessgroup.com

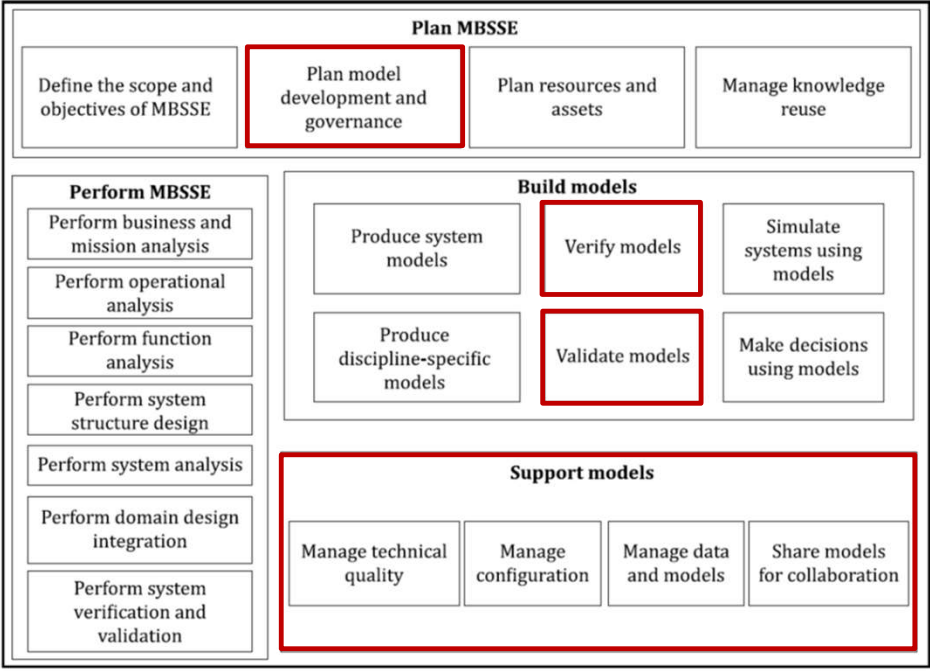
About APG

- APG mission

“Align architecture and engineering capabilities with business strategy using proven, practical processes delivering world-class results.”

- Industry thought leader in enterprise architecture, business architecture, systems and software engineering, business analysis, and agile methods
- Member and contributor to
 - UML[®], SysML[®], UAF[®] at the Object Management Group
 - TOGAF[®], ArchiMate[®], IT4IT[™] at The Open Group
 - Architecture, MBSE, and OOSEM Workgroups at INCOSE

Continuous MBSE Modeling



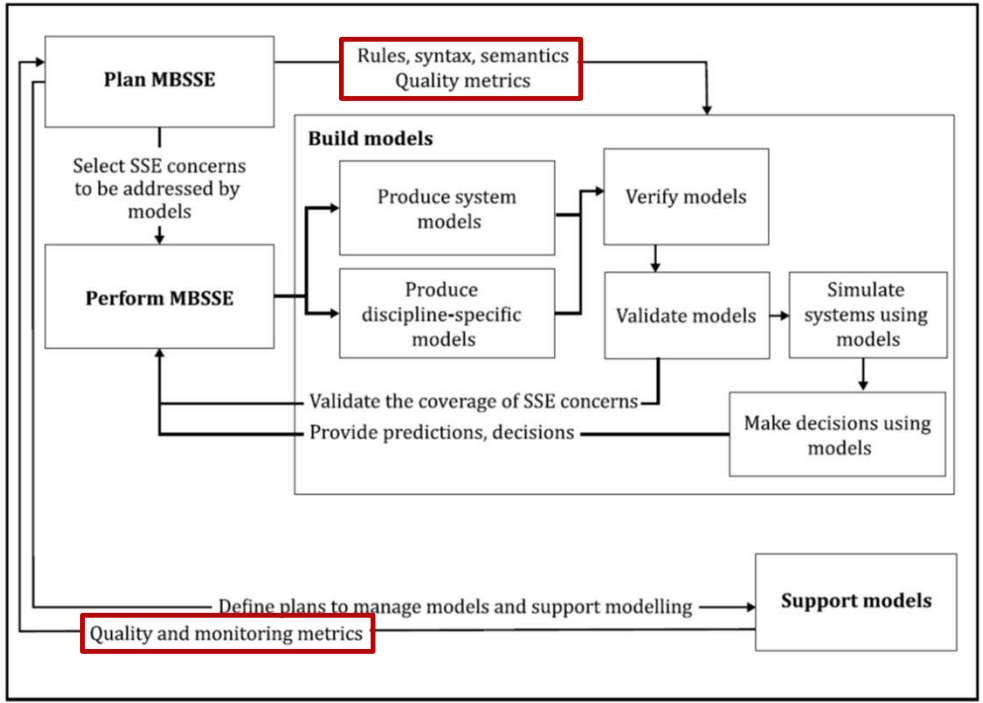
Modeling Tools



Sparx Enterprise Architect



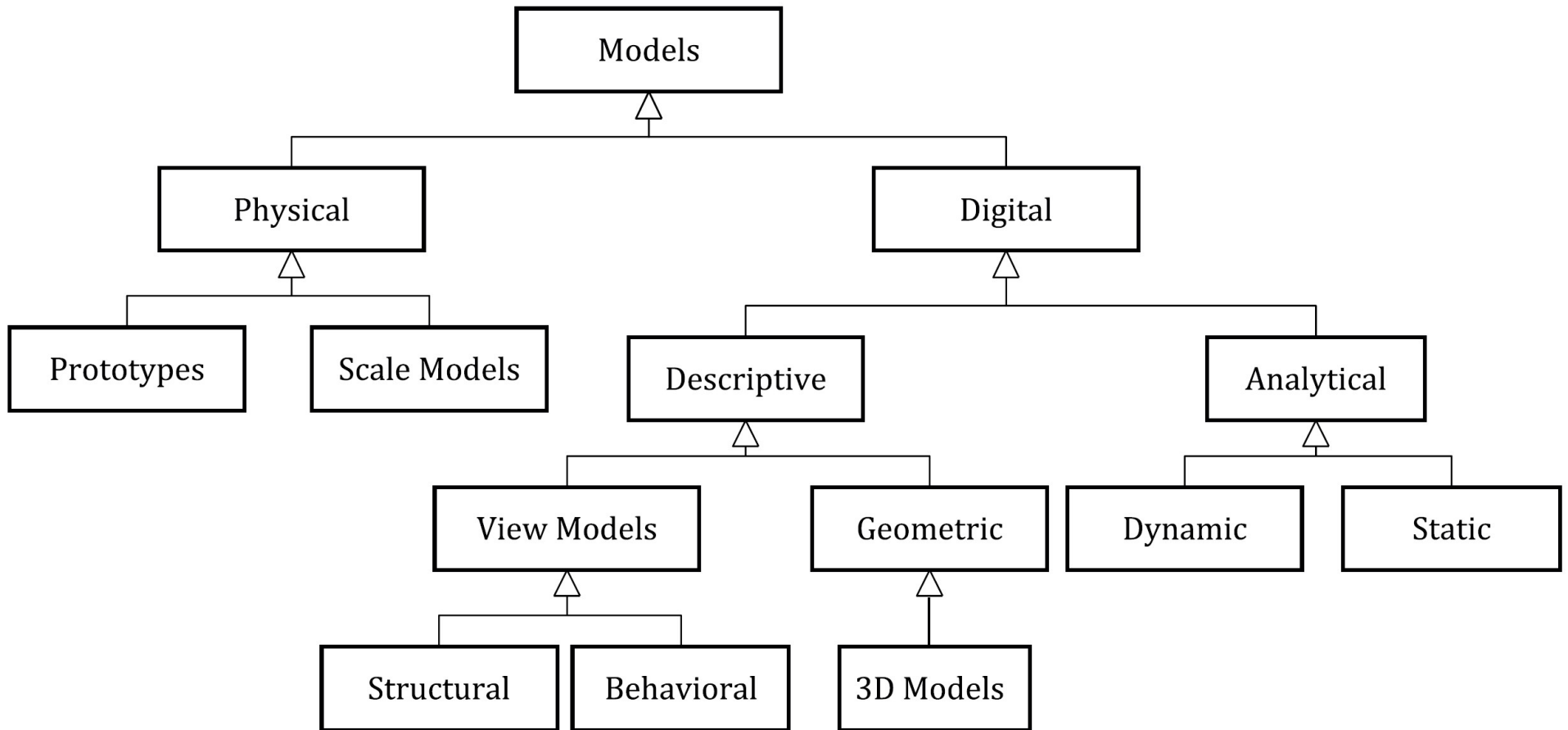
IBM Rational Rhapsody



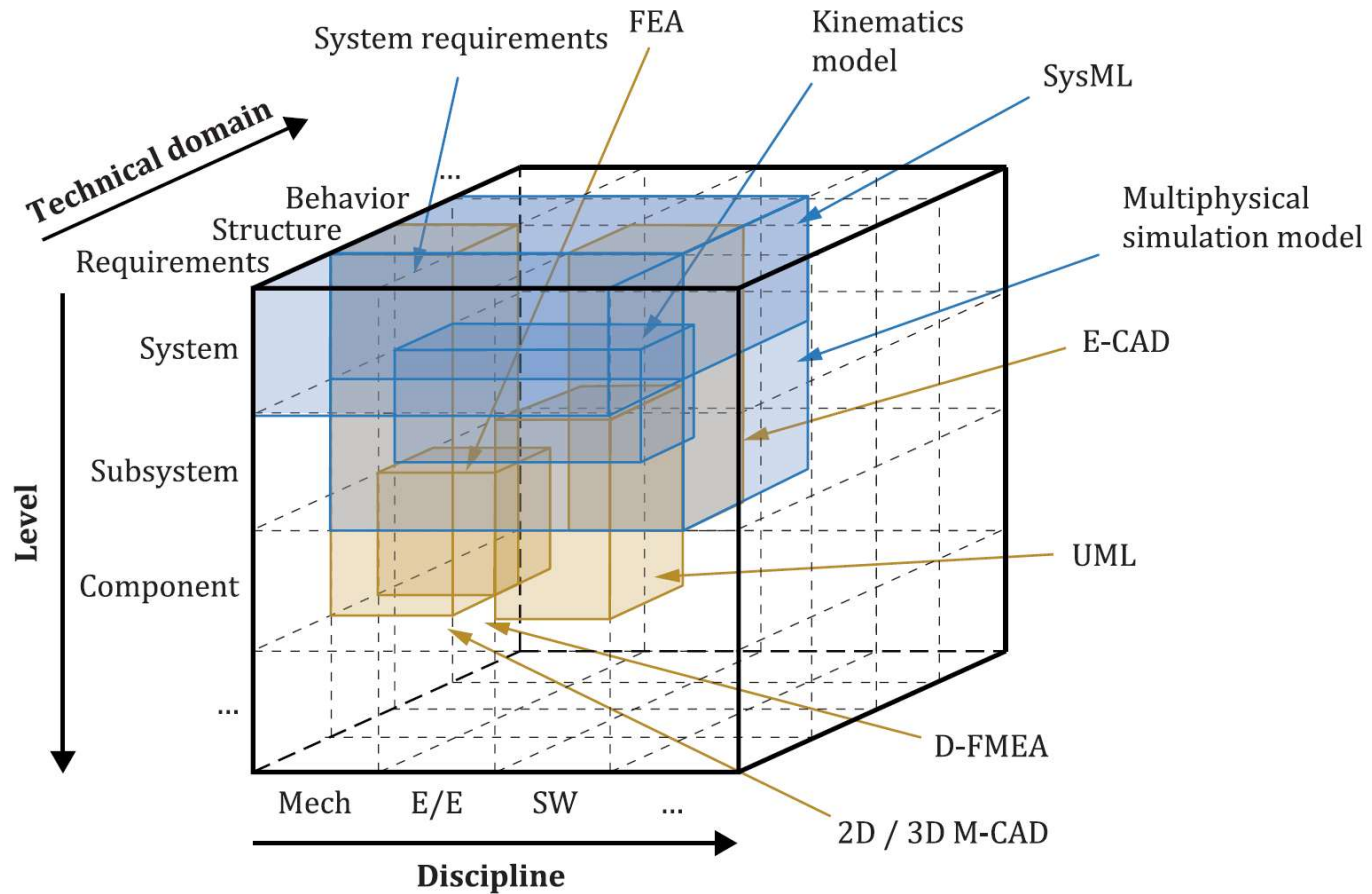
CI/CD Pipeline Automation



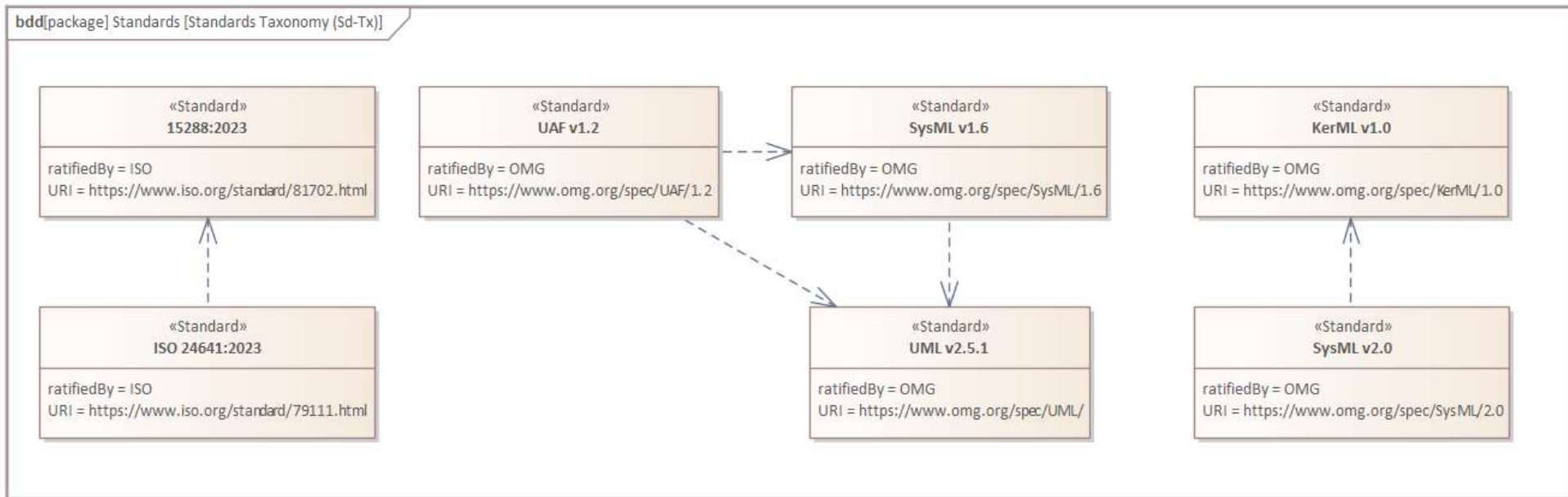
Types of Models



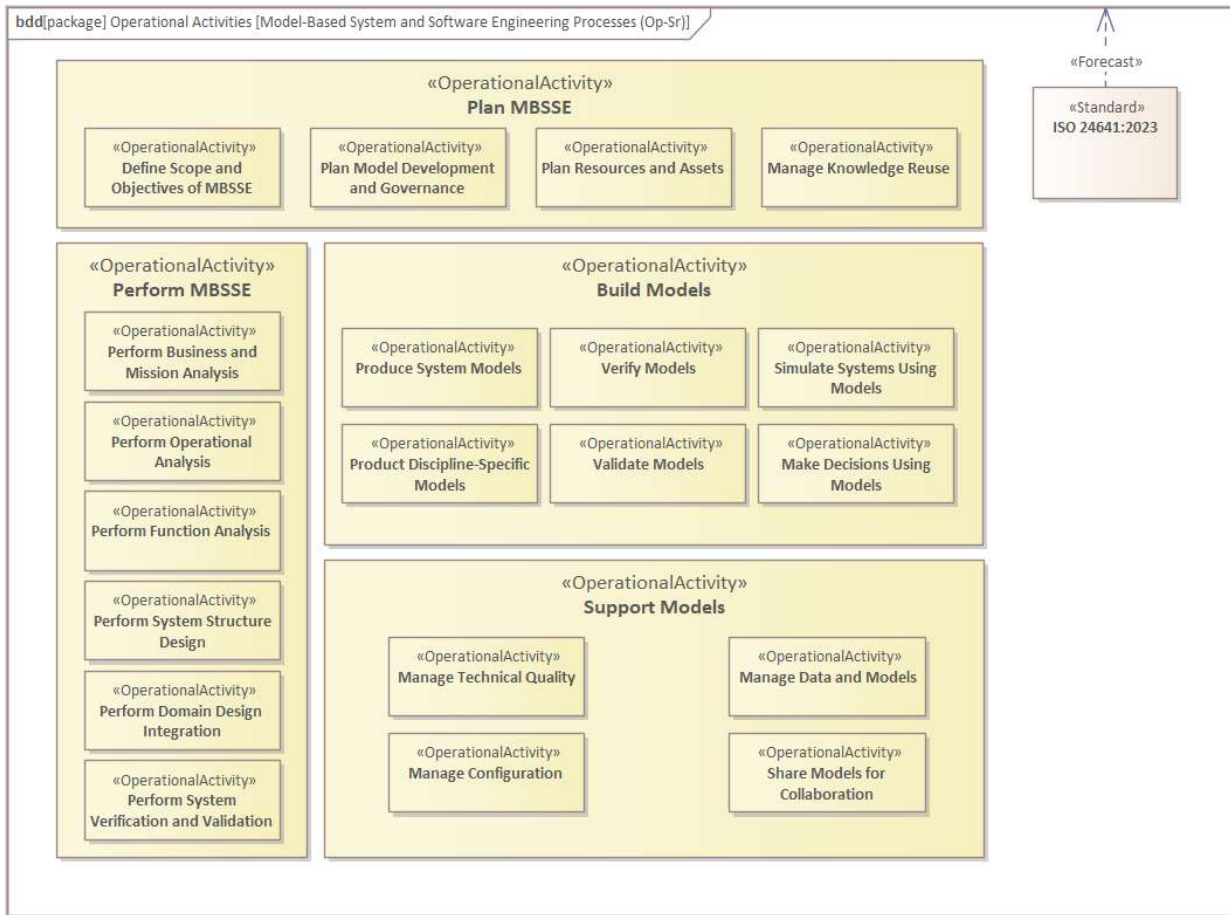
System Model Dimensions



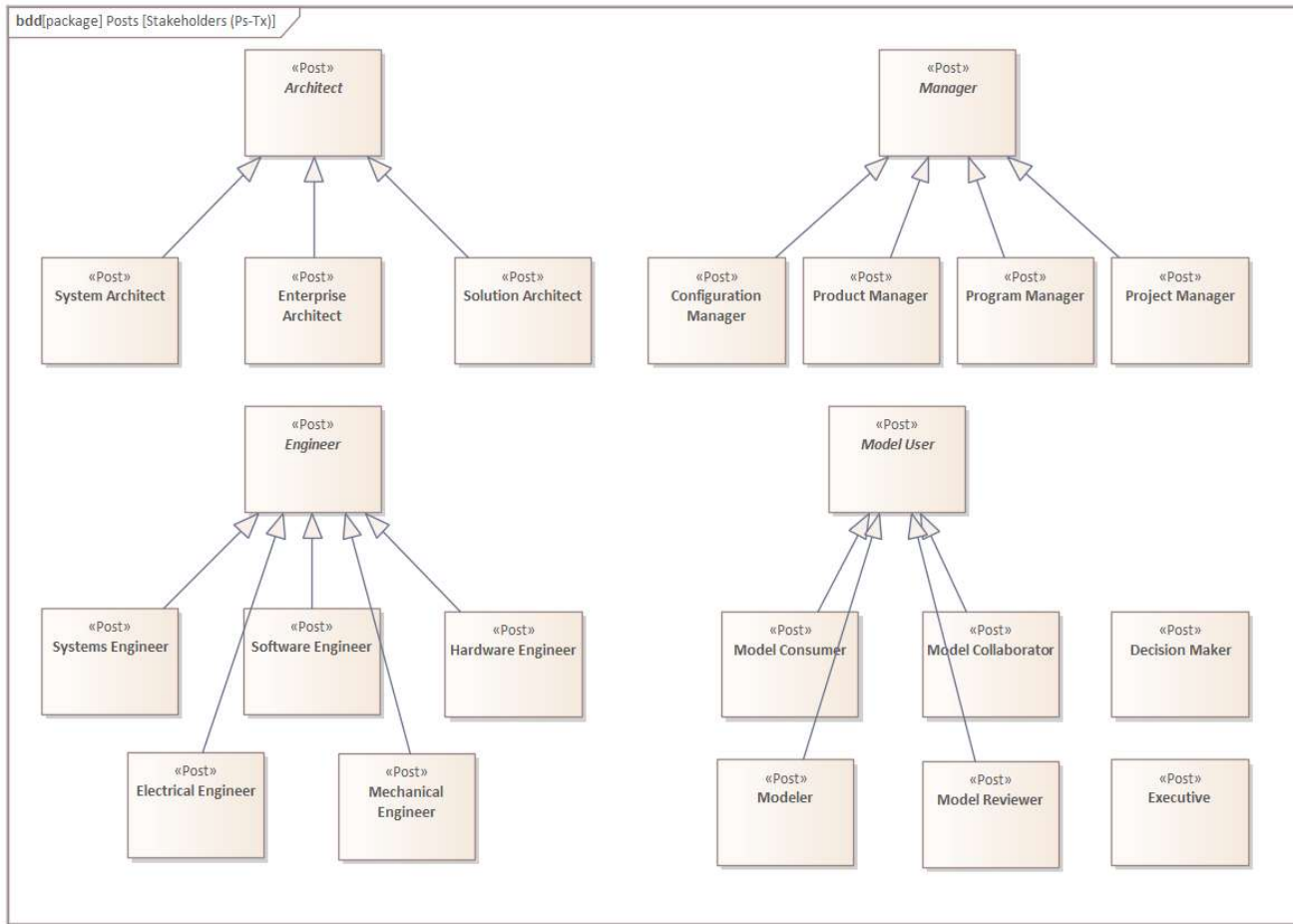
Standards Taxonomy (Sd-Tx)



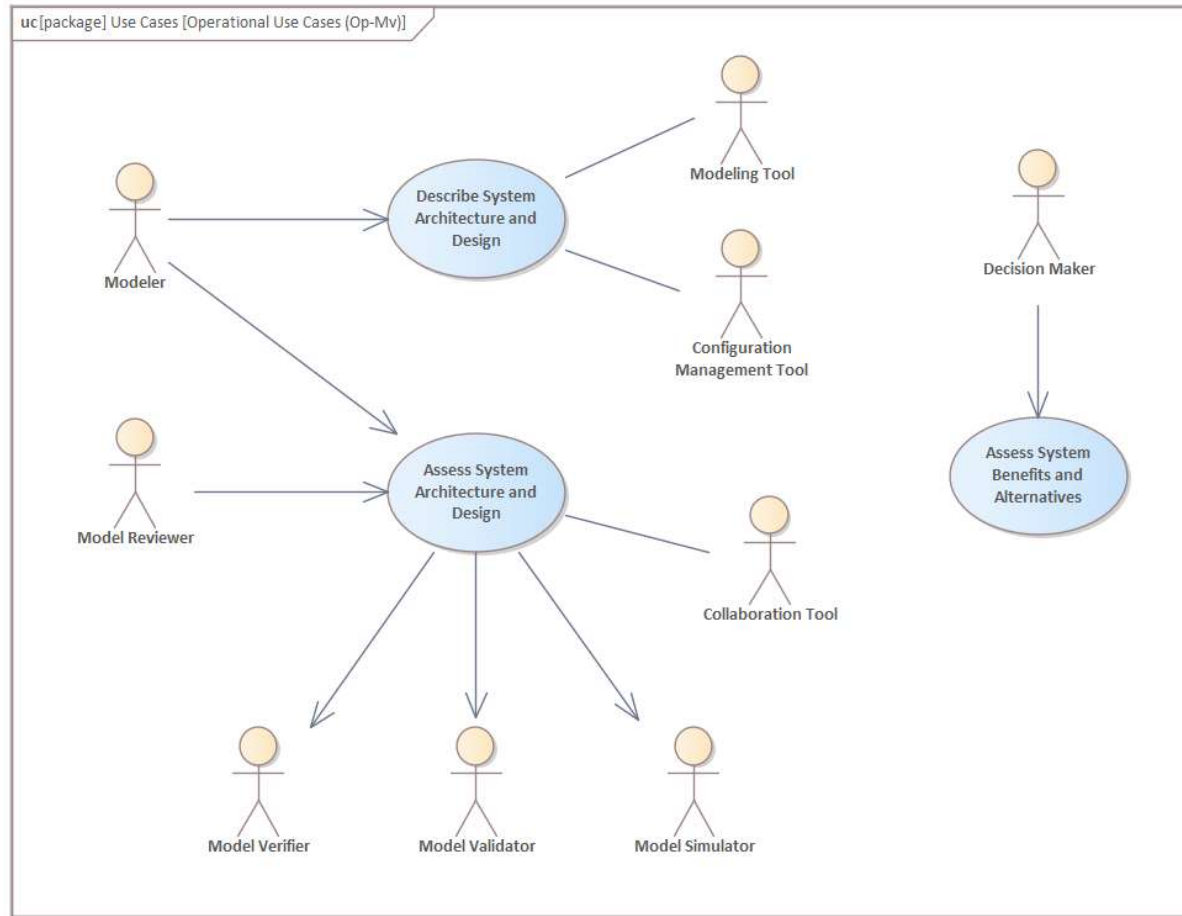
MBSSE Processes (Op-Sr)



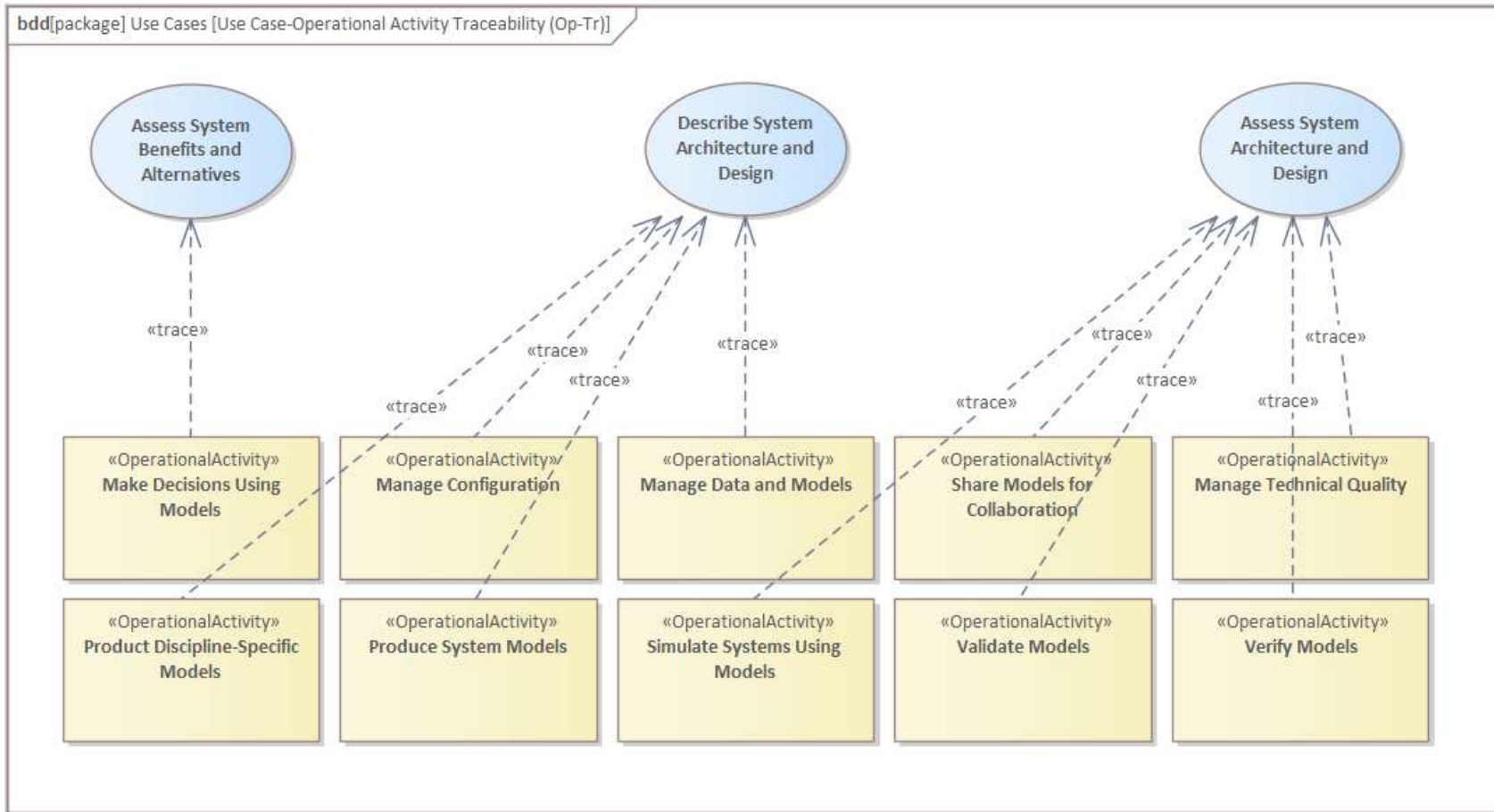
Stakeholders (Ps-Tx)



Operational Use Cases (Op-Mv)



Use Case-Operational Activity Traceability (Op-Tr)



Plan Model Development and Governance



Produce Models

bdd[package] Verify Models [Verify Models Activities (Op-Pr)]

«OperationalActivity»
Maintain Model Consistency

«OperationalActivity»
Maintain Model Integrity

«OperationalActivity»
Evaluate Model Conformance

«OperationalActivity»
Assess Model Level of Detail
and Accuracy

bdd[package] Validate Models [Validate Models Activities (Op-Pr)]

«OperationalActivity»
Ensure Models Meet
Expectations

«OperationalActivity»
Enable Model Isolation

«OperationalActivity»
Enable Model
Composition/Federation

«OperationalActivity»
Enable Model Usage for
Simulation and Analysis

Support Models

bdd[package] Manage Technical Quality [Manage Technical Quality Activities (Op-Pr)]

«OperationalActivity»
Perform Technical Review

«OperationalActivity»
Perform Quality Assurance

bdd[package] Manage Data and Models [Manage Data and Models Activities (Op-Pr)]

«OperationalActivity»
Define Modeling
Management Policy

«OperationalActivity»
Define Data/Model
Management Infrastructure

bdd[package] Manage Configuration [Manage Configuration Activities (Op-Pr)]

«OperationalActivity»
Manage Modeling Assets
and Configuration Items

«OperationalActivity»
Manage Model Changes

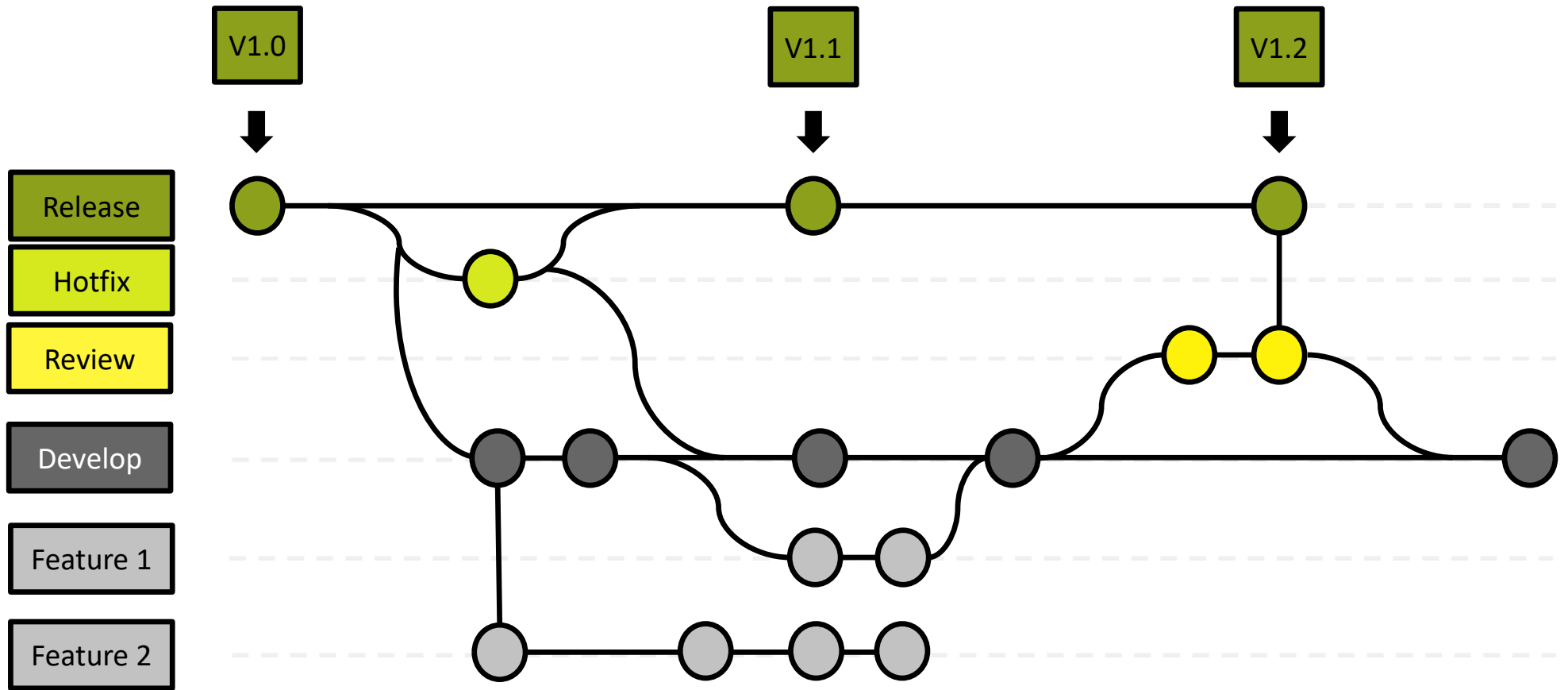
bdd[package] Share Models for Collaboration [Share Models for Collaboration Activities (Op-Pr)]

«OperationalActivity»
Define Collaborative
Modeling Guidelines and
Environment

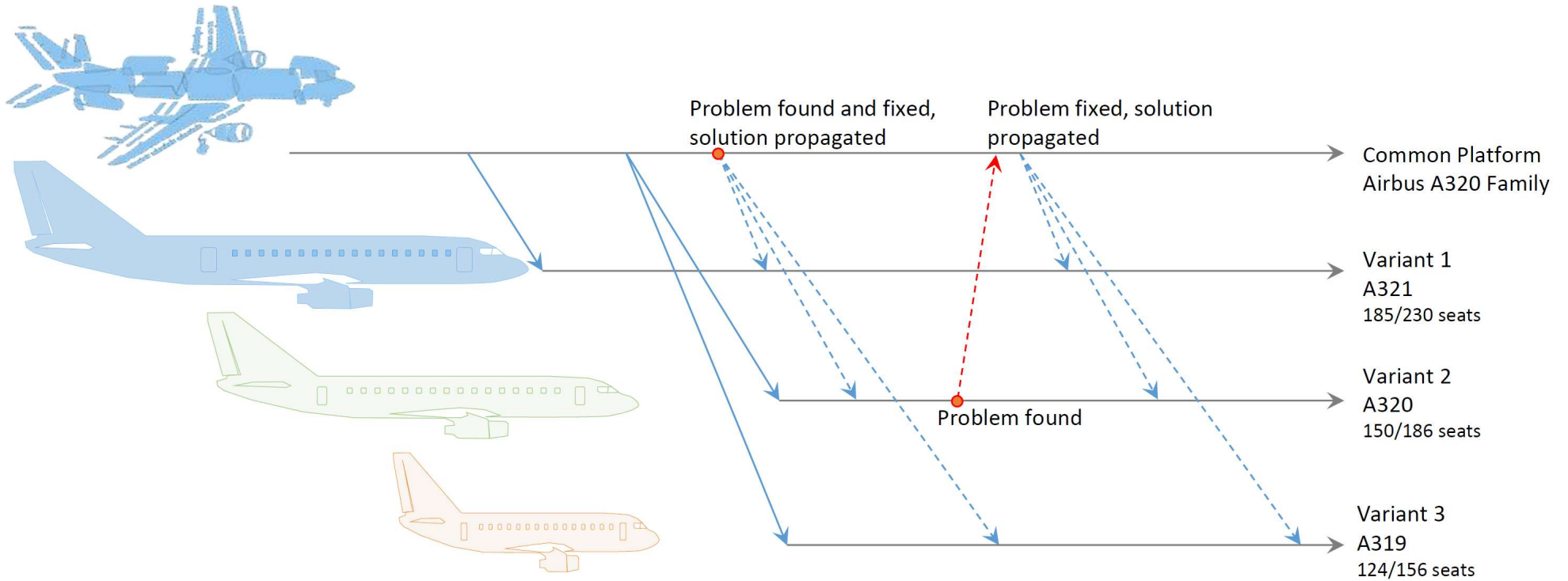
«OperationalActivity»
Define Model Sharing and
Authoring Rules

«OperationalActivity»
Maintain Model Consistency

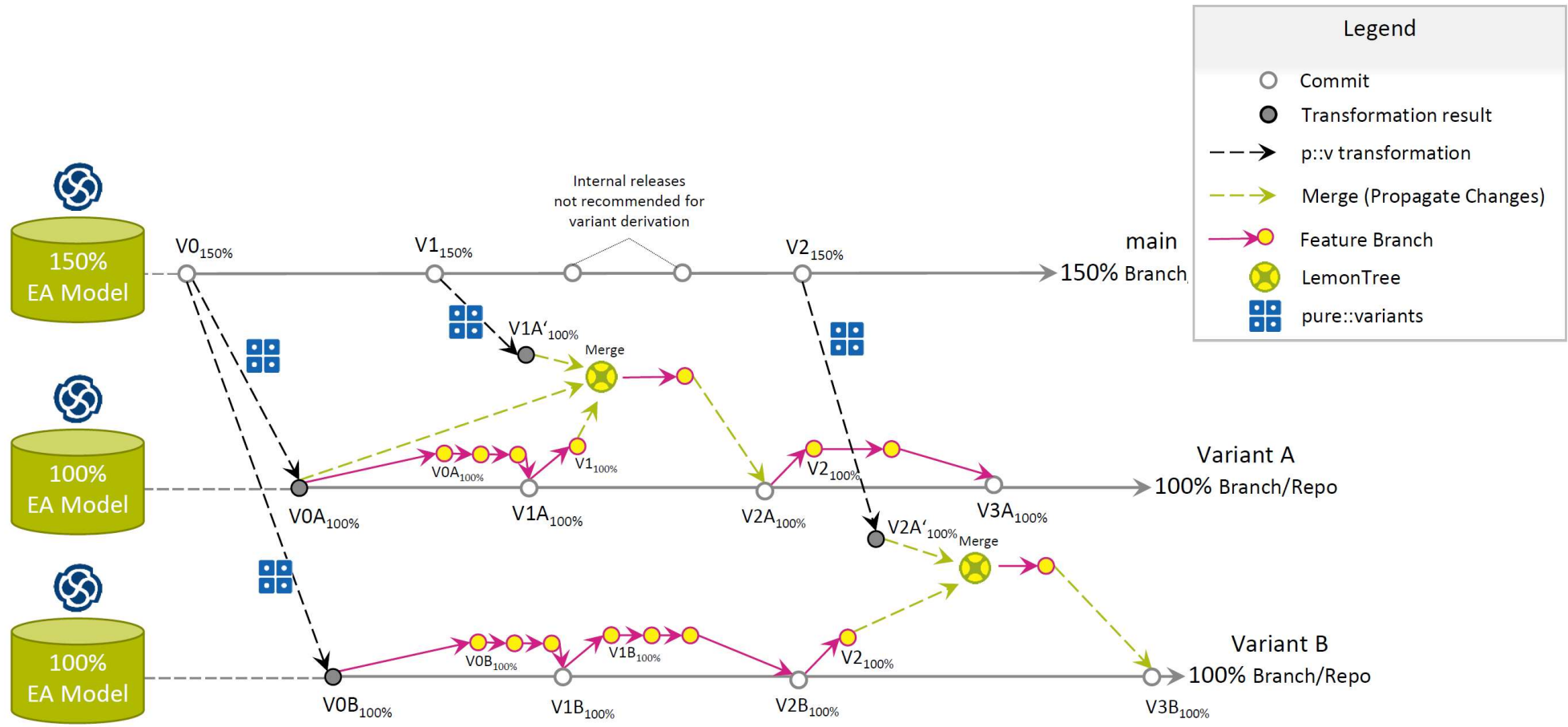
Gitflow Branching Strategy



Variant Management



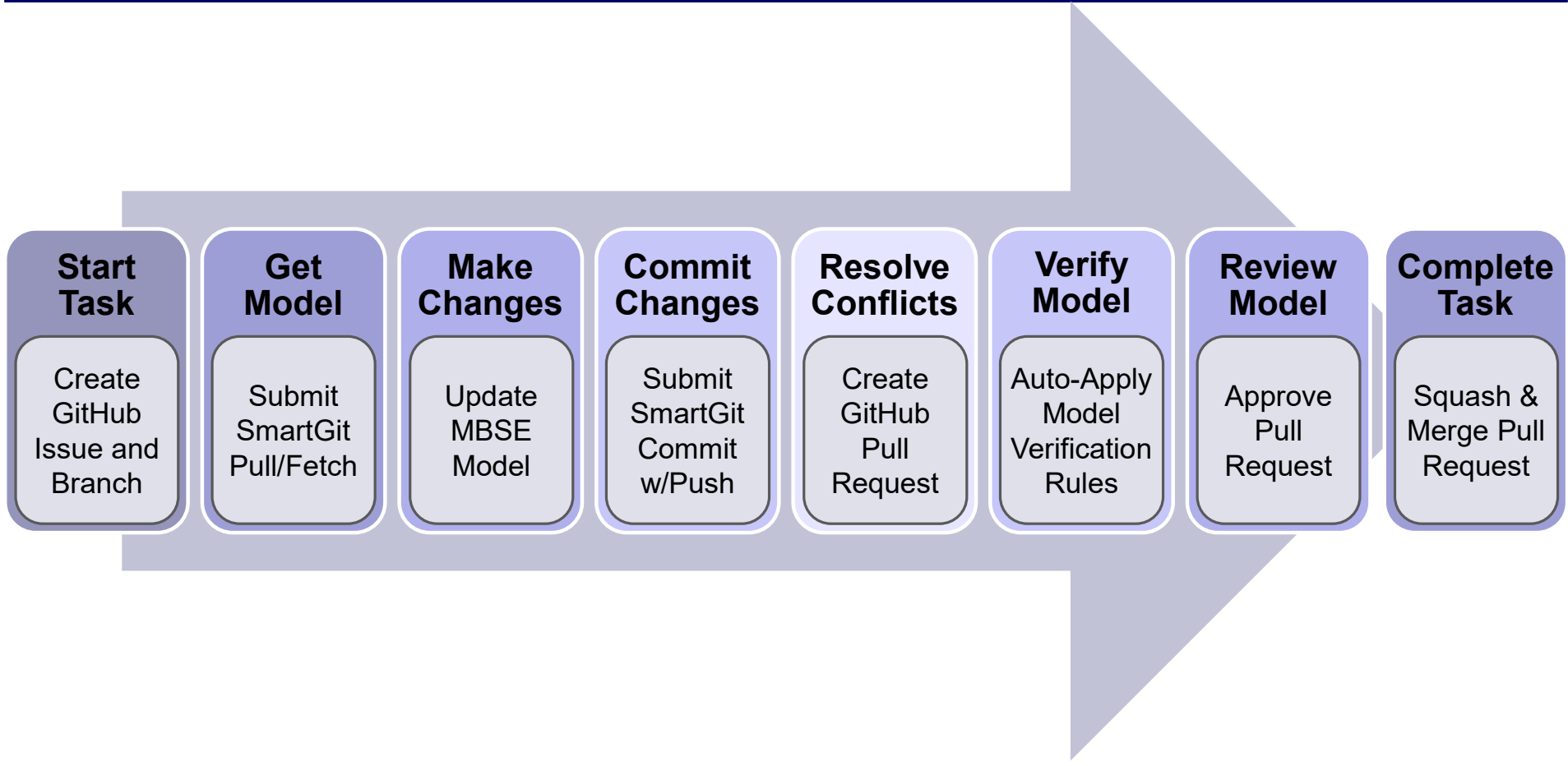
Versions and Variants



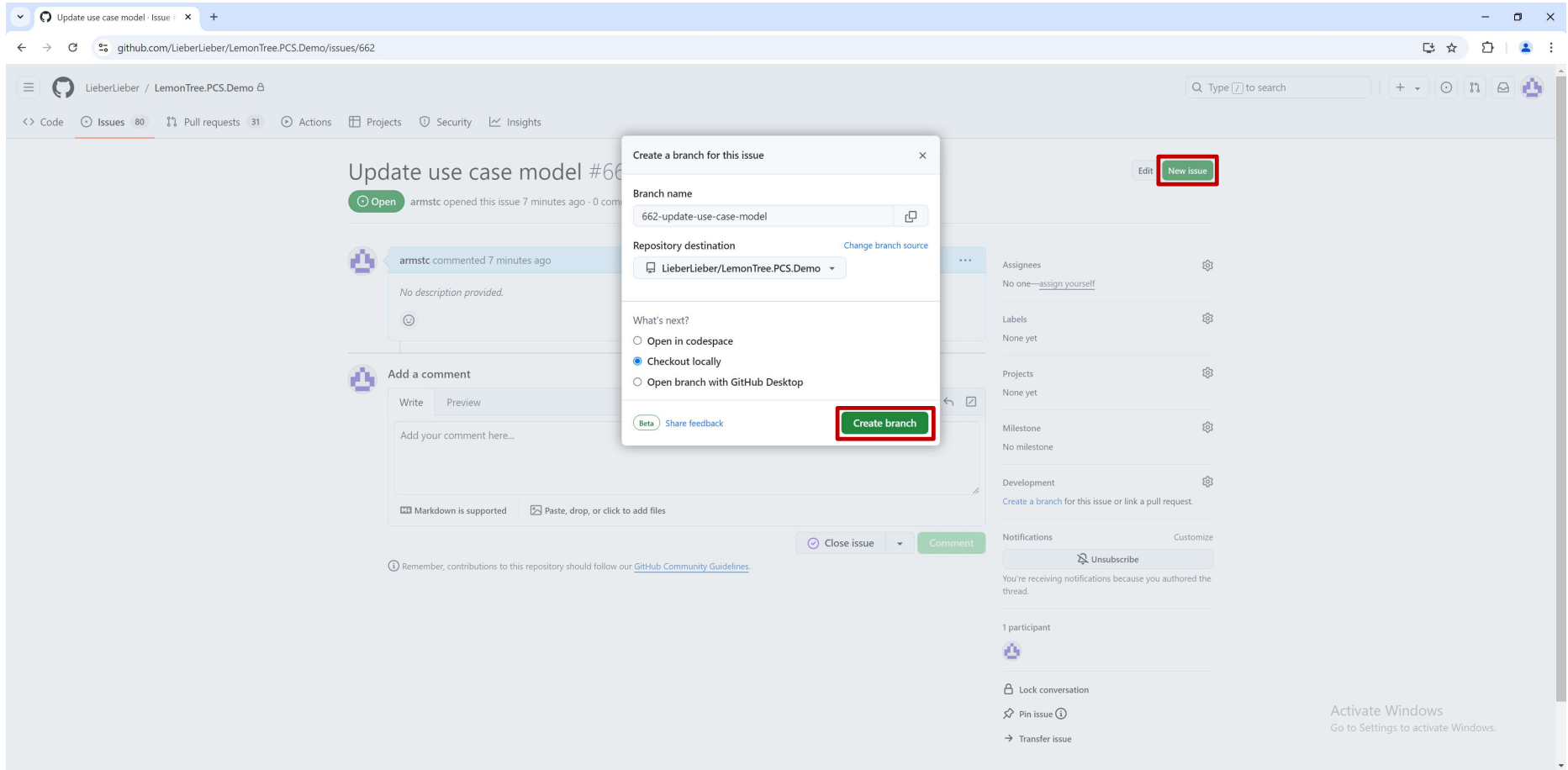
CI/CD Principles

Principle	MBSE Perspective
Repeatable Reliable Process	<ul style="list-style-type: none"> • Consistency of process across lifecycle and environments is critical • Leverage proven software engineering practices for MBSE
Automate Everything	<ul style="list-style-type: none"> • Humans need to build models (although AI might assist) • Automate model merging, conflict resolution, verification, and reviews
Version Control	<ul style="list-style-type: none"> • Critical for regulatory certification/auditing and feature-based MBSE • Three-way merging essential for parallel development
Build In Quality	<ul style="list-style-type: none"> • Ensure projects are following organizational processes and modelers are using modeling languages appropriately
Do Hard Parts First	<ul style="list-style-type: none"> • Tackle complexity of MBSE modeling lifecycle as soon as possible • Manage technical debt associated with MBSE models
Everyone Is Responsible	<ul style="list-style-type: none"> • Provide accessibility and visibility of modeling activities to all team members
“Done” Means Released	<ul style="list-style-type: none"> • Define and enforce criteria for releasing MBSE models and derived artifacts
Continuous Improvement	<ul style="list-style-type: none"> • Difficult to improve something that is not defined, controlled, and measured

Approach – GitHub Flow



Start Task: Create GitHub Issue and Branch



Get Model: Submit SmartGit Pull/Fetch

The screenshot displays the SmartGit application interface for a repository named 'LemonTree.PCS.Demo'. The main window shows a pull request for the branch 'origin/662-update-use-case-model' against the 'main' branch. The pull request details include the commit hash '712b8a7a' by Chris Armstrong, dated 10/11/2024 at 02:21 PM. The commit message is '652 get local copy of model (#657)'. The pull request is currently in a 'Pending' state. The interface includes a 'My History' sidebar on the left, a central commit history view, and a right-hand pane showing the commit details and a list of changed files, including 'Model.qeax'. The bottom status bar indicates 'Command succeeded'.

Make Changes: Update MBSE Model

The screenshot shows the Enterprise Architect interface with a use case diagram. The diagram on the left shows a 'User' actor connected to three use cases: 'Merge Model', 'Diff Model', and 'Validate Model'. A list of tasks is overlaid on the diagram:

- Delete “Merge Model” use case
- Rename “Diff Model” to “Compare Model”
- Add new “Demo Integration” use case

The diagram on the right shows the updated state: the 'User' actor is now connected to 'Compare Model' and 'Validate Model', and a new 'Demo Integration' use case has been added. Below the diagrams is a 'Validation Results' table:

Severity	Category	Rule	Description
ⓘ	Requirements	0163_REQUIREMENTVERIFY	Requirement: Integrate with modern version control systems is not verified.

At the bottom of the screenshot, the status bar shows: 'uc [package] Use Cases [Demo Use Cases]: created: 10/10/2024 11:57:30 AM modified: 10/11/2024 5:15:21 PM 125% 650 x 1098'.

Commit Changes: Submit SmartGit Commit w/Push

The screenshot displays the SmartGit application interface. At the top, the menu bar includes options like Repository, Edit, View, Remote, Local, Branch, Query, Tools, Window, and Help. Below the menu is a toolbar with icons for Pull, Push, Start, Integrate, Finish, Cherry-Pick, Revert, and Apply Stash. The main workspace is divided into three panes: 'My History' on the left, 'All branches' in the center, and a commit details pane on the right. The 'My History' pane shows a list of commits, with the most recent one highlighted: '662-update-use-case-model' (CA) with the message 'Added new use case #662'. The 'All branches' pane shows a commit graph with 'origin' and 'main' branches. The right pane shows the commit details for 'Added new use case #662', including the commit hash '31e0806a' and the author 'Chris Armstrong' with the date '10/11/2024 05:19 PM'. A red box highlights this commit information. At the bottom, a file comparison pane shows 'Model.qeax' with a message: 'File size exceeds configured limit of 1,000,000 bytes.' The bottom status bar indicates 'Command succeeded.'

Resolve Issues: Review Model Changes

The screenshot displays a GitHub pull request interface on the left and the LemonTree comparison tool on the right. The pull request, titled "Update use case model #662", shows a commit from "armstc" with a message indicating "No merge conflicts in model". Below the commit message, a diagram titled "[package] Use Cases" is shown, featuring a stick figure labeled "User" connected to three ovals: "Compare Model", "Validate Model", and "Demonstrate Integration".

The LemonTree tool is in "Compare" mode, comparing "Model_base.qeax" and "Model_head.qeax". The tool's interface includes a search bar, a list of impacted elements (4 / 4), and a tree view of the model structure. The "Demo Use Cases" diagram is highlighted, showing the same elements as the pull request. The tool also displays "Impacted Diagrams [2 / 2]" and "Element Properties" for the selected diagram, listing graph edges such as "[unnamed] - { User, Diff Model }" and "[unnamed] - { User, Merge Model }".

Verify Model: Auto-Apply Model Validation Rules

Demo/pull/663

Review use case changes #662 #663
armstc wants to merge 1 commit into main from 662-update-use-case-model

github-actions bot commented 7 minutes ago

IncQuery Validator for Enterprise Architect

Analysis were executed on Model.qeax with the SAIC Digital Engineering Validation ruleset containing 91 rules.

The following type of issues were detected:

Level	Findings
Error	3
Info	1

Out of the 91 rules, 4 have findings.

✖ The model failed the validation criteria, there should be no errors or fatal errors.

To view the full report download the SAIC Digital Engineering Validation analysis result artifacts from [here](#).

github-actions bot commented 7 minutes ago

[Validation Report](#)

Some checks were not successful
1 failing and 3 successful checks

- ✖ Verify IncQuery Validator / IncQueryValidatorForEA (pull_request) Failing after 1m
- ✔ LTA publish review session to PR / PublishReviewSession (pull_request_target) Successful in 46s
- ✔ LTA update review session to PR's after commit to main / build (push) Successful in 7s
- ✔ Verify Modelcheck / ModelReadinessCheck (push) Successful in 30s

This branch has no conflicts with the base branch

Validation Report

Overview

Severity	Rules with findings	Findings
Error	3 rules	3/81 (4%)
Info	1 rule	1/1 (100%)

Rules without findings: No findings out of 493 possible for 19 rules

Inapplicable rules: 68 inapplicable rules

Rules with findings

Category	Name	Description	Severity	Findings
Completeness	0208_UCDOC UMENTATION	All activities, actors, constraint blocks, operations, signals, states, stereotypes (with their public properties) and use cases must have documentation. (Activities that are methods for operations or are classifier behaviors for use cases are exempt.)	Error	1/27 (4%)
Use Case Integrity	0206_UCACT OR	Use cases not connected to other use cases via extend/include/generalization relationships must be associated with at least one actor (or actor subtype).	Error	1/27 (4%)

Complete Task: Squash & Merge Pull Request

The screenshot displays a GitHub pull request for 'Review use case changes #662'. The main content area shows the results of a validation analysis performed on 'Model.qeax' using the 'SAIC Digital Engineering Validation' ruleset. A table indicates one finding of 'Info' level. Below this, a comment from 'github-actions' provides a link to the 'Validation Report'. To the right, a UML Use Case diagram titled 'uc[package] Use Cases [Demo Use Cases]' shows a 'User' actor associated with 'Compare Model', 'Validate Model', and 'Demonstrate Integration' use cases. A dashed '«trace»' relationship connects 'Demonstrate Integration' to an '«interfaceRequirement» Integrate with modern version control systems' requirement, which is noted as being 'from Requirements'. At the bottom, a 'Checks' section shows four successful checks: 'All checks have passed', 'LTA publish review session to PR', 'LTA update review session to PR's after commit to main', and 'Verify IncQuery Validator'. A 'Squash and merge' button is highlighted with a red box.

- Added association from actor to new use case
- Added documentation to new use case
- Traced new use case to requirement

Activate Windows
Go to Settings to activate Windows.

Refresh Local Repository

The screenshot shows the SmartGit 24.1 RC 3 interface. The top menu bar includes Repository, Edit, View, Remote, Local, Branch, Query, Tools, Window, and Help. The main toolbar contains icons for Local Files, History, Pull v, Push, Start, Integrate, Finish, Cherry-Pick, Revert, and Apply Stash v. The interface is divided into three main sections: My History, All branches, and a diff view.

My History: Shows the current branch 'main' and a list of recent commits, including 'Review use case changes #662 (#663)'. The commit message for this commit is highlighted in a red box.

All branches: Shows a list of branches and their corresponding commit hashes and dates.

Diff View: Shows the changes in the file 'Model.qeax'. The diff content is: 'File size exceeds configured limit of 1,000,000 bytes.' The diff view includes options for Force Compare, Unified, Side by Side, Compact, and Ignore WS.

Commit History (Highlighted):

- Commit: 37ac0624
- by armstc, 10/11/2024 05:51 PM
- committed by GitHub
- Review use case changes #662 (#663)
- * Added new use case #662
- * Fixed model issues #662

SysMLv2 Model Example

Base

```

1 package Car{
2   part def Engine{
3     part cylinder;
4   }
5
6   part def Chassis{
7     part wheel;
8     part def Breaksystem{
9       part break;
10      part tractionControl;
11    }
12  }
13 }

```

A

```

1 package Car{
2   part def Engine{
3     part cylinder;
4   }
5
6   part def Chassis{
7     part wheel;
8     part def Breaksystem{
9       part diskBreak;
10      part tractionControl;
11    }
12  }
13 }

```

B

```

1 package Car{
2   part def Engine{
3     part cylinder;
4   }
5
6   part def Chassis{
7     part wheel;
8     part def Breaksystem{
9       part drumBreak;
10
11      part tractionSystem;
12    }
13  }
14
15  part def body{
16    part door;
17    part trunk;
18    part windshield;
19  }
20 }

```

SysMLv2 Model Merging

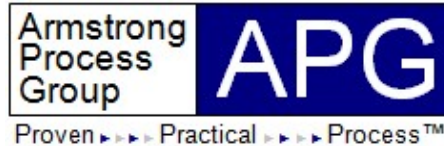
The screenshot displays the LemonTree software interface for SysMLv2 model merging. The main workspace is divided into three panels: 'compare_Car-32b6c81d0ec4b6...', 'compare_Car-889c6366a7480945d0bfa00...', and 'Merge Target'. The 'Merge Target' panel shows a 'SysMLv2 Model (A)' with a 'Car (A)' component containing a 'Chassis (A)', 'Breaksystem (A)', and 'body (B)'. The 'Breaksystem (A)' contains a 'drumBreak (A)', '[Element not existing] (B)', and 'tractionSystem (B)'. The 'body (B)' contains 'door (B)', 'trunk (B)', and 'windshield (B)'. The 'Element Properties' panel at the bottom shows a conflict for the 'Name' property of 'breardrumBreak', with options 'Take A' and 'Take B'. The 'Take B' option is selected, and the value is 'NEITHER ONE OF THOSE'. The 'Impacted Elements' panel on the left shows a list of elements with their status: 'diskBreak' (Modified), 'tractionControl' (Removed), and 'tractionSystem' (New).

Conclusions

- ISO/IEC 24641 provides best practices for developing, managing, and leveraging models for systems engineering
- CI/CD is an accepted best practice in software engineering
 - Key to sustainable agile development
 - While ISO/IEC 24641 doesn't explicitly mention CI/CD, some of its requirements can be satisfied by CI/CD adoption
- CI/CD is natural evolution towards next level of MBSE performance and capability

Q&A

31



<http://www.aprocessgroup.com>

chris.armstrong@aprocessgroup.com

***Thanks for your attention
and participation!***

"APG", the APG logo, "proven practical process" (and its graphic representation) are trademarks of Armstrong Process Group, Inc.

Object Management Group, OMG, Systems Modeling Language, SysML, Unified Architecture Framework, UAF, Unified Modeling Language, and UML are trademarks or registered trademarks of the Object Management Group, Inc. in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Continuous MBSE Modeling

Copyright © 1998-2026 Armstrong Process Group, Inc., All rights reserved

APG