Collaborating with OpenMBEE as an Authoritative Source of Truth Environment

Benjamin Kruse, Sc.D.

Systems Engineering Research Center

Systems Engineering Research Center (SERC)

Collaborating with OpenMBEE as an Authoritative Source of Truth Environment

By:
Benjamin Kruse, Sc.D.
Mark Blackburn, Ph.D.

SYSTEMS ENGINEERING
Research Center

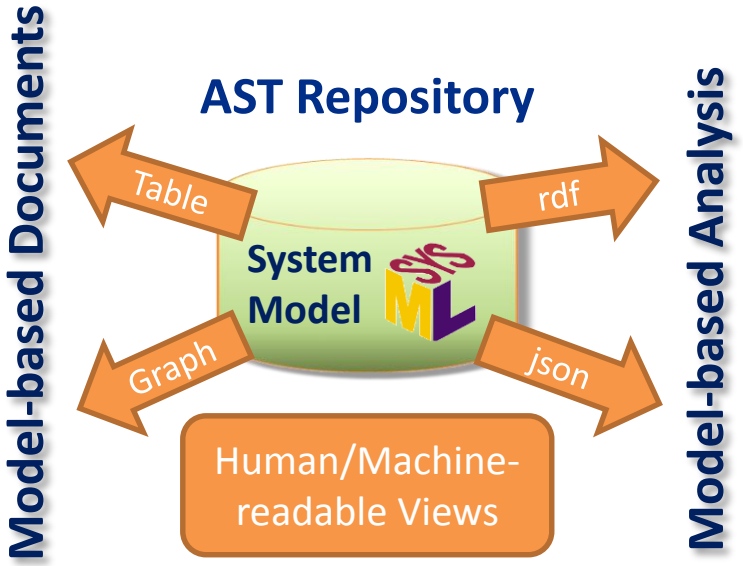
Copyright and Disclaimer

Agenda

- Authoritative Source of Truth & Surrogate Pilot
- OpenMBEE Overview
  - Model Development Kit (MDK) & DocGen
  - Model Management System (MMS)
  - View Editor
- Results
  - Digital Signoff Mechanism
  - Guidelines and Findings
    - Project Usage & User Permissions Example
  - Issues & Suggested Improvements

Introduction

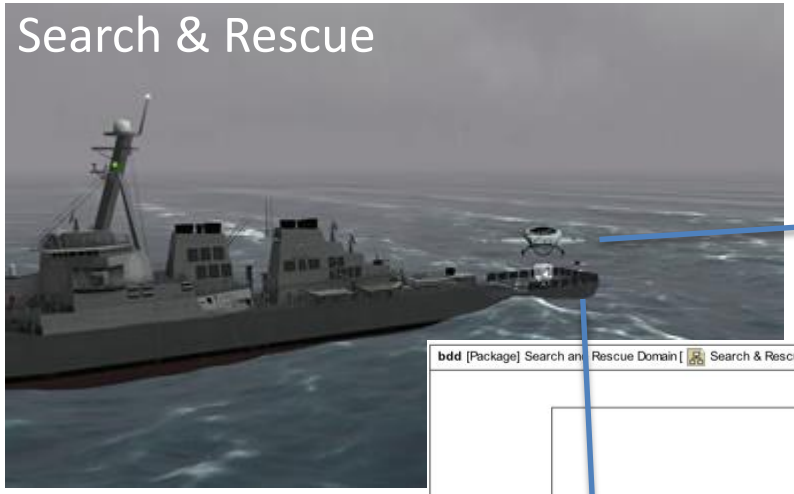
- Authoritative Source of Truth (AST)
  - To provide consistent data in the format necessary for the given task
  - Implemented by **OpenMBEE** = Open Model Based Engineering Environment, developed by NASA/JPL



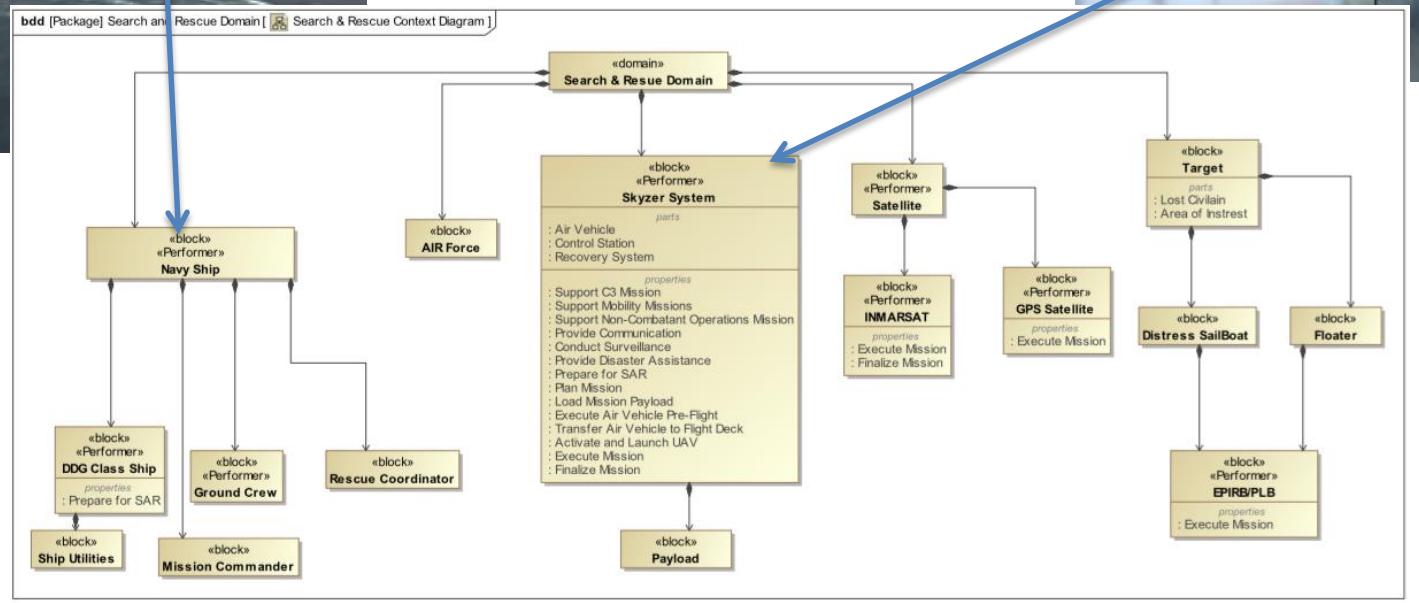
- Surrogate Pilot
  - Execution of NAVAIR's Systems Engineering Transformation (SET) Framework
  - To simulate collaboration in an AST
  - To support new operational paradigm between government and industry
    - By elimination of paper artifacts and large-scale design reviews in favor of continuous insight/oversight via the digital collaborative environment

Surrogate Pilot Scenario: Skyzer UAS

Graphical CONOPS Scenario:



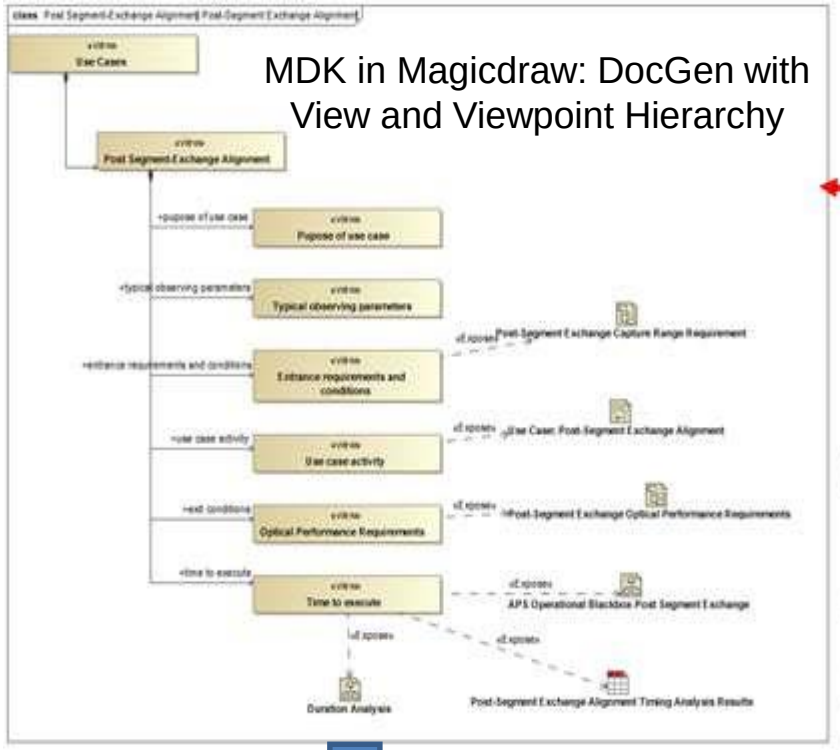
Airfoil designation for a similar Bell tilt-rotor (BA609)



Skyzer System & Mission Models developed using SysML

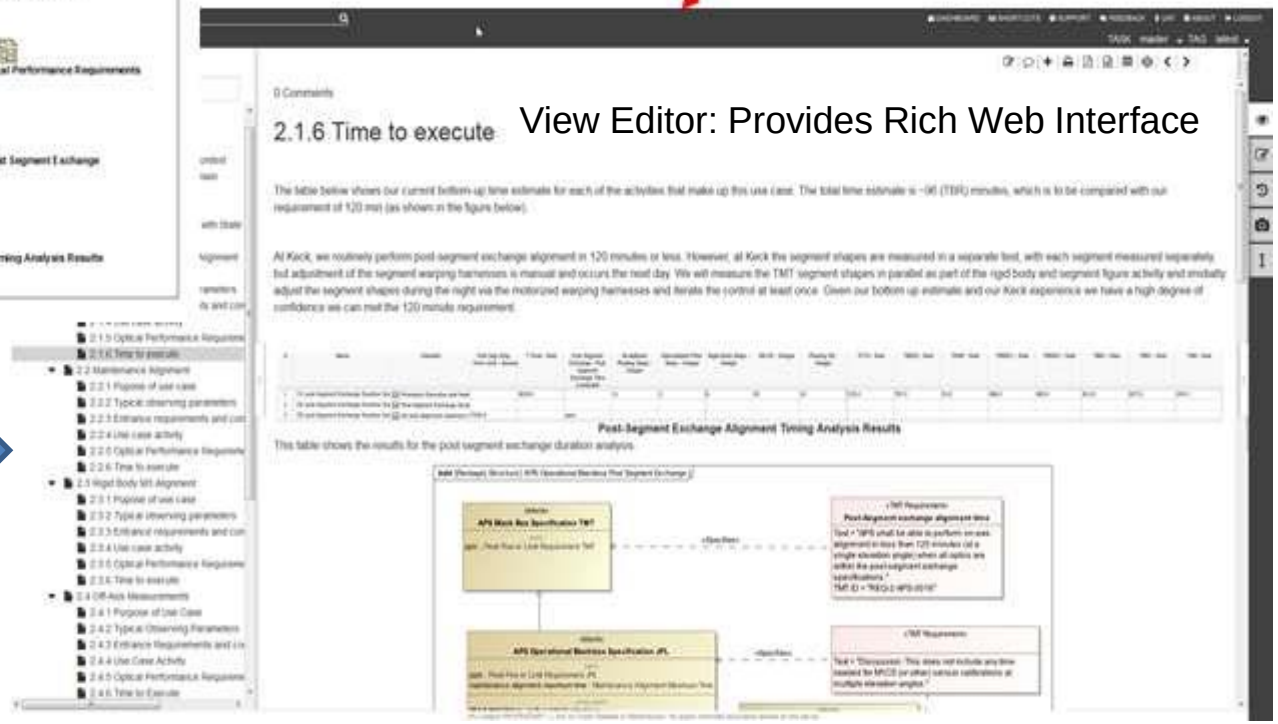
OpenMBEE: MDK, MMS & View Editor

MDK in Magicdraw: DocGen with View and Viewpoint Hierarchy



Model Management System (MMS)

Visualization in View Editor

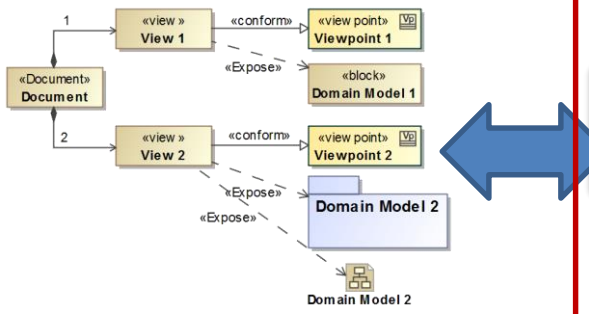


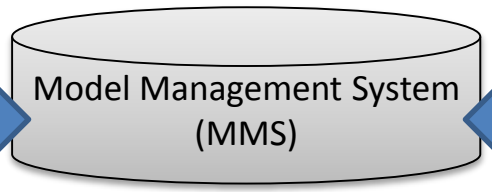
View Editor: Provides Rich Web Interface

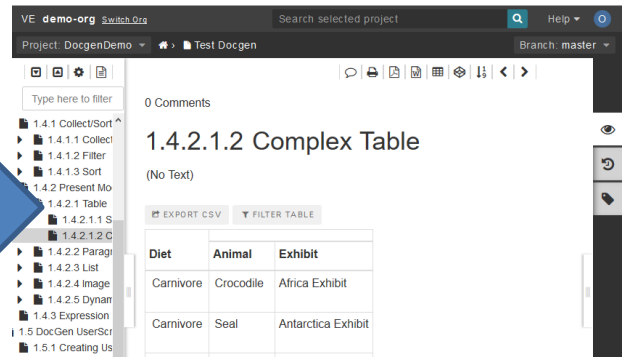
Model Development Kit (MDK)

- What is MDK?
  - Plugin for Magicdraw, to support building system assemblies through modeling augmentation and validation, enable syncing with MMS and using the DocGen language for model-based document creation using views and viewpoints
  - Content
    - Systems Reasoner
    - MMS Sync
    - DocGen

MDK: View and Viewpoint Hierarchy



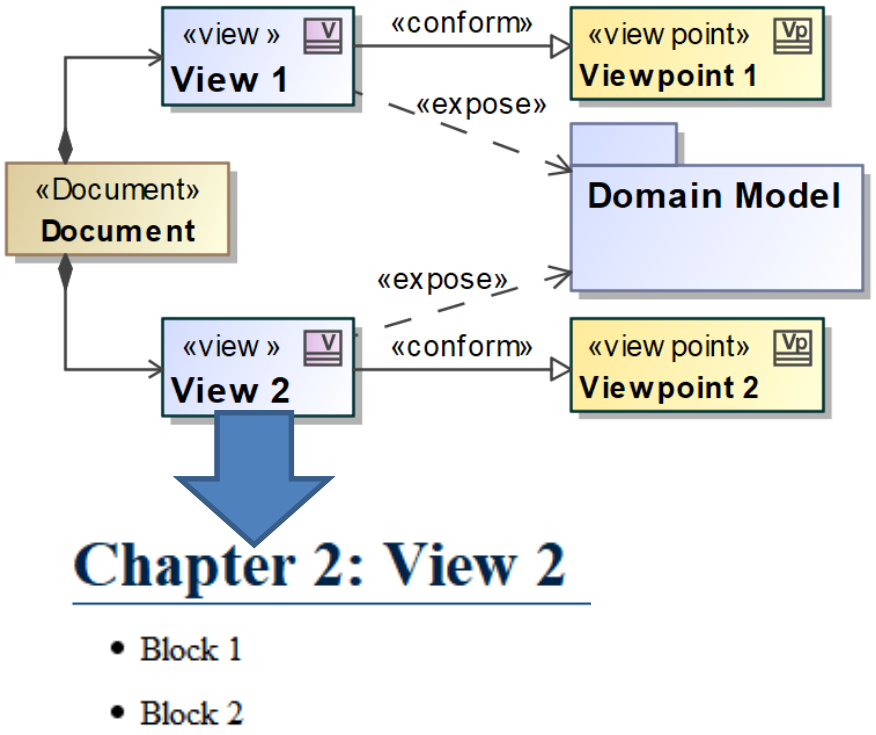


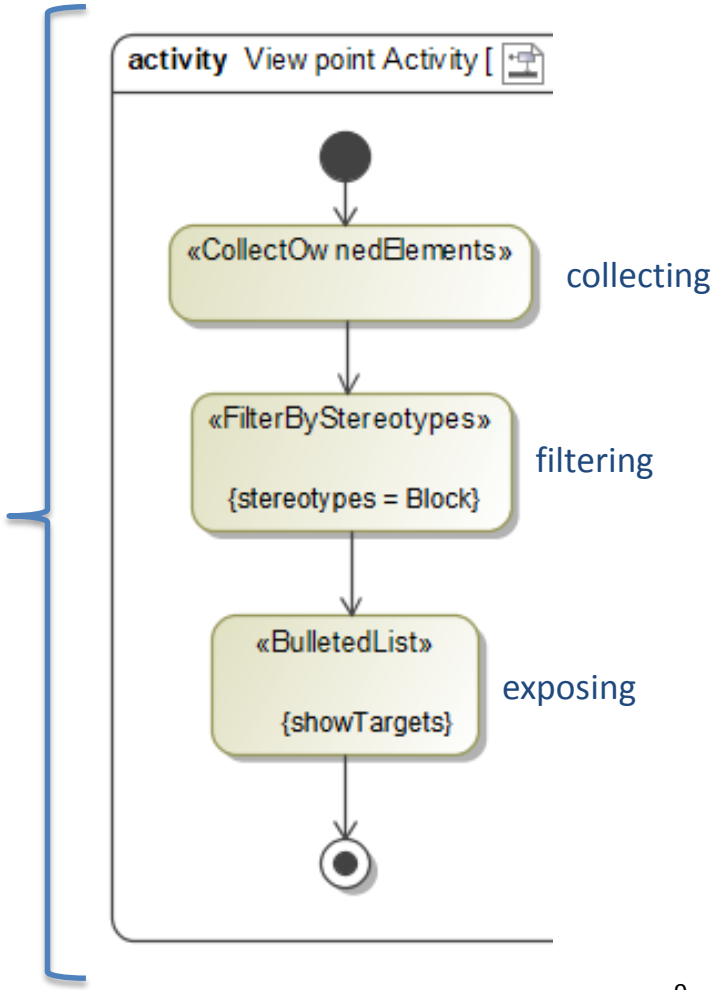


View Editor: Provides Rich Web Interface

MDK: DocGen

- DocGen
  - For model-based document creation based on Views and Viewpoints
    - To generate views for View Editor or pdfs
    - To guide modeling and development



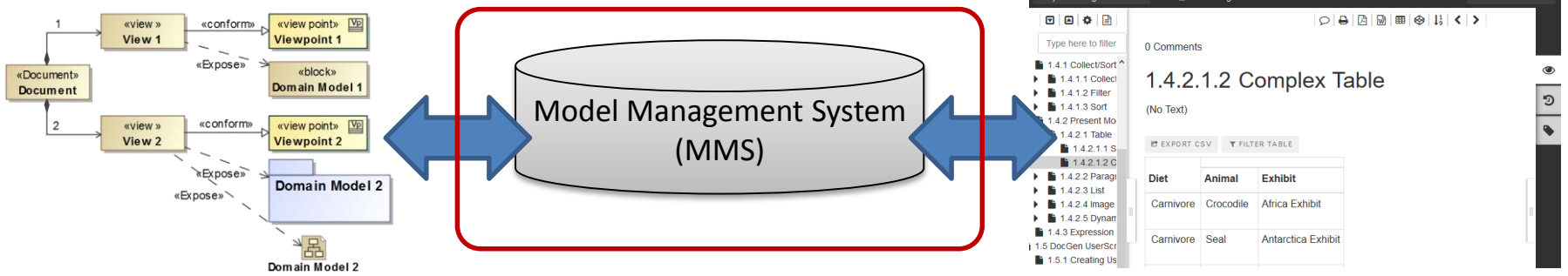


Model Management System (MMS)

- What is MMS?
  - —A version control system for structured data, including versioning, workflow management, and controlled access through RESTful web services
  - —Used as central data hub to facilitate multi-tool and multi-repository integration across engineering, computing, and management disciplines

- To store SysML model data
  - —Capturing all model elements (e.g. classes, instances, relations, but not: diagram layout), including their change history and views for View Editor

MDK: View and Viewpoint Hierarchy

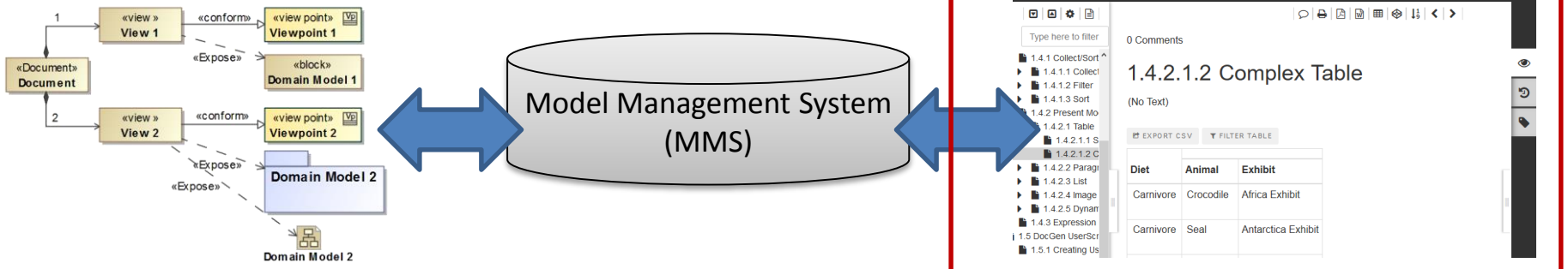


View Editor: Provides Rich Web Interface

View Editor

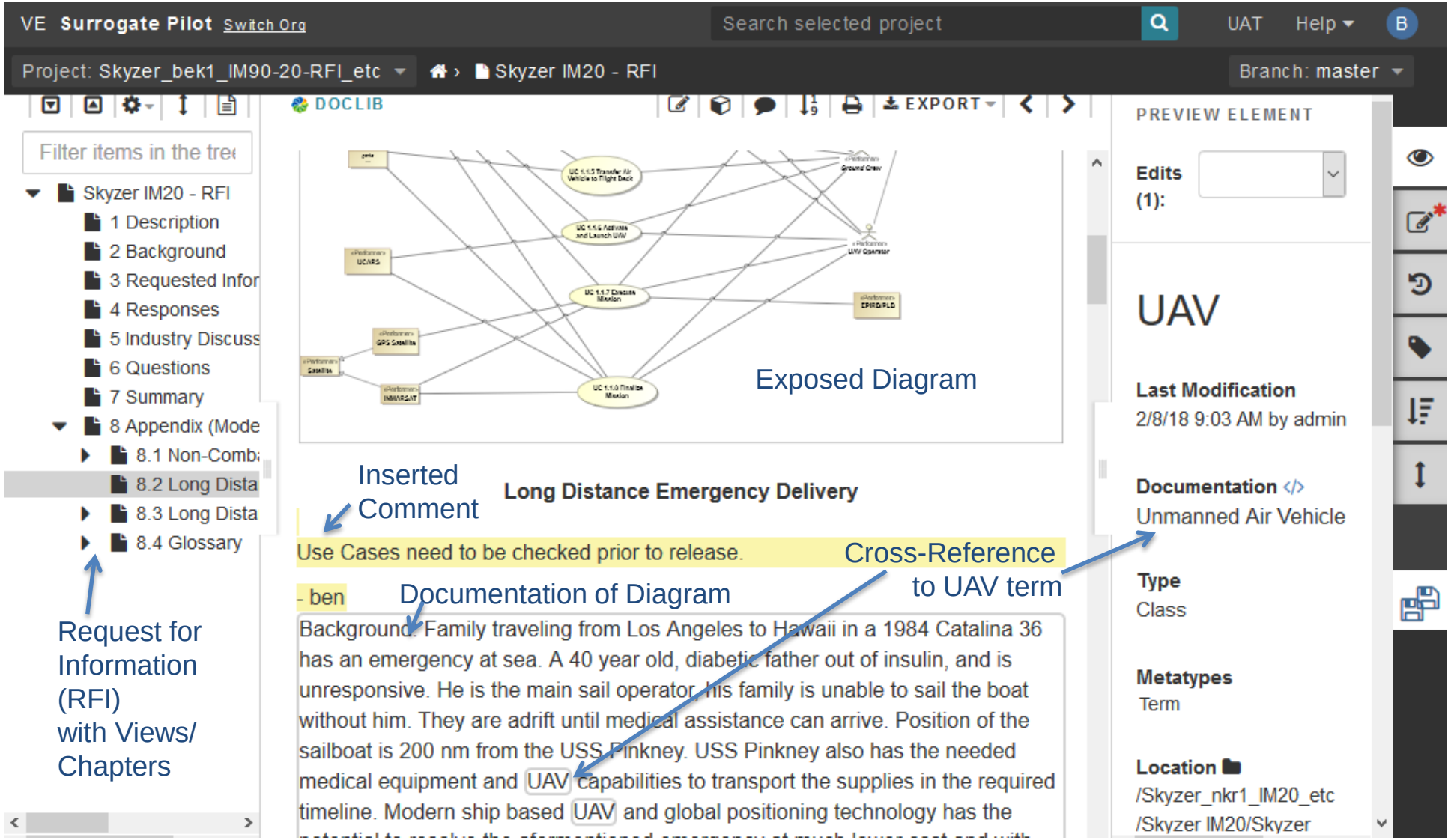
- What is the View Editor?
  - —A web app to provide consistent DocGen views on live SysML data, to allow interaction with model elements outside of the case tool
- To communicate through and edit model-derived documents
  - —Provide AST data access to non-modelers
  - —Editing of exposed model elements (e.g. placeholders for new elements) and addition of presentation elements (e.g. text, videos, etc.) and comments

MDK: View and Viewpoint Hierarchy

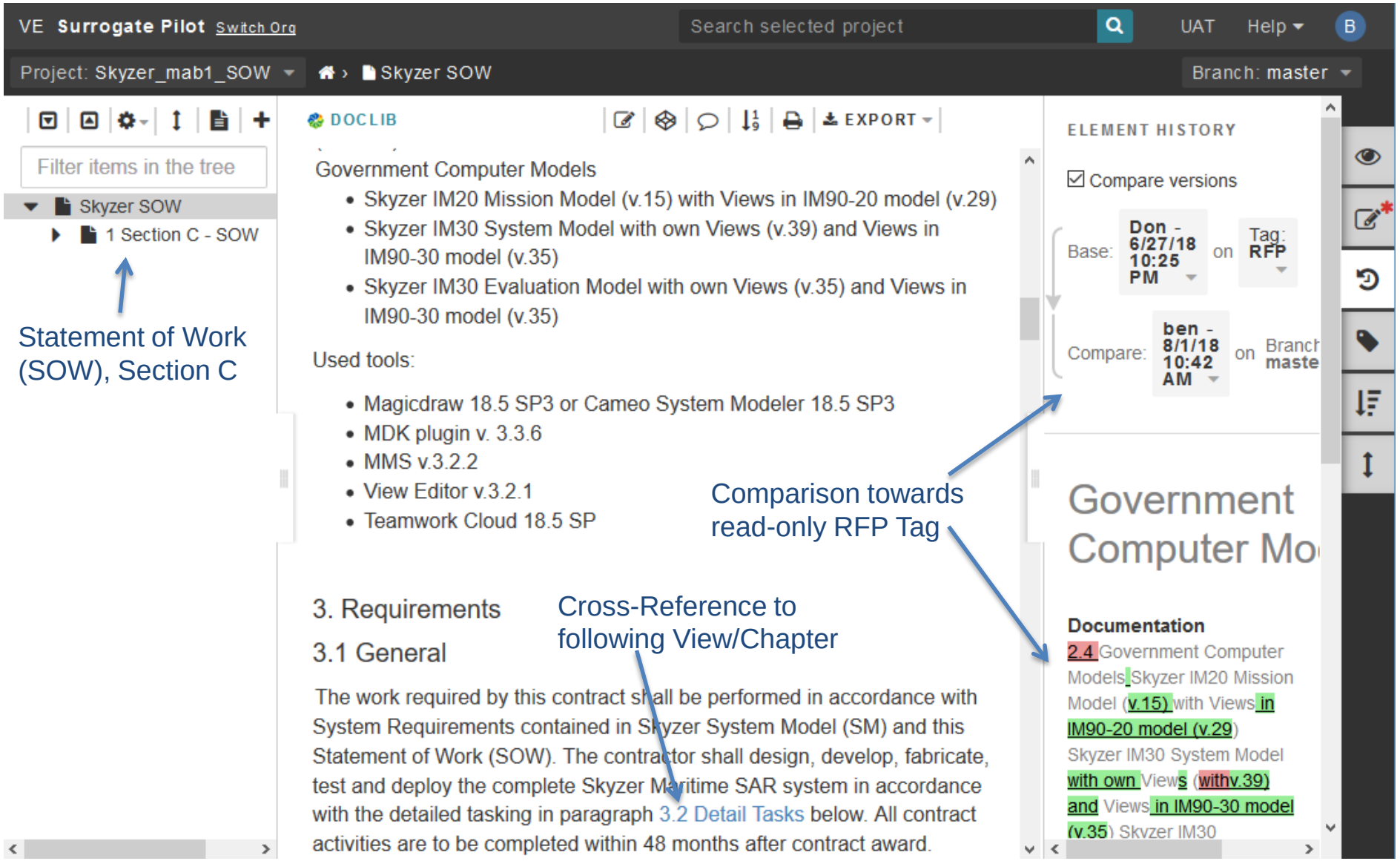


View Editor: Provides Rich Web Interface

View Editor: Commenting & Cross-Reference



View Editor: History Comparison

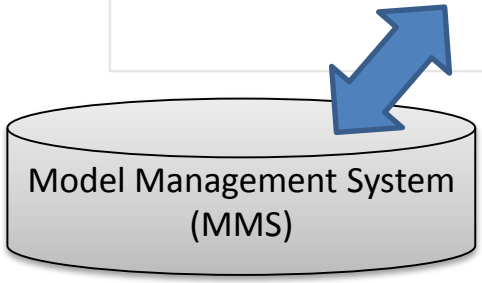


Digital Signoff

Digital Signoff for cross-referenced elements in View Editor, to support transition from document-based to model-based documents

General Performance SignOff

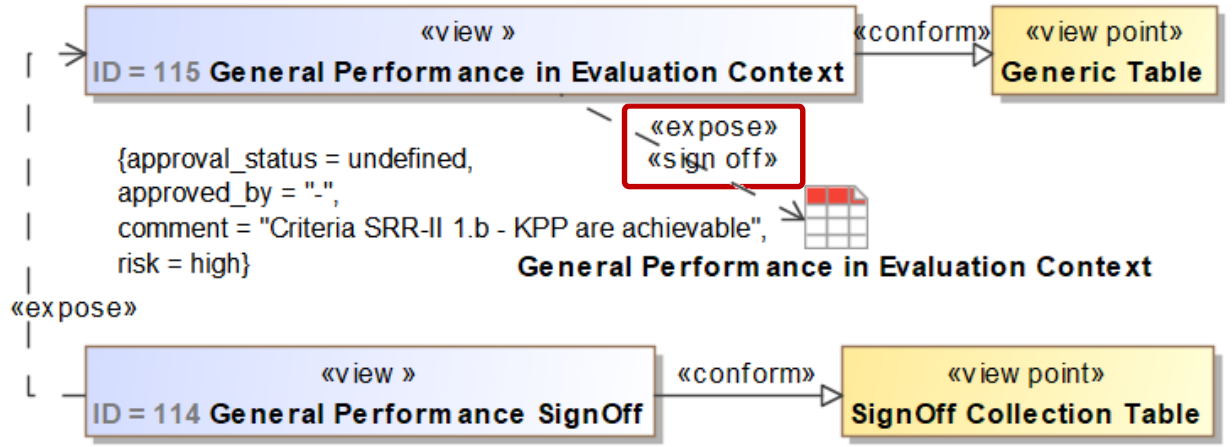
Approved Elements	Risk	Approval Status	Approved By	Comment
General Performance in Evaluation Context	high	Value : undefined (to be defined / undefined / approved / rejected)	-	Criteria SRR-II 1.b - KPP are achievable



Signoff target: General…
Approval status: undefined
Last changed by: Admin
Last changed at:
02/22/2019, 10:43:51

Captured signoff & model information in AST (including who and when)

Specification of model elements to be signed off in DocGen view hierarchy



Guidelines and Findings (1)

- View Editor offering views on consistent data from AST, improving communication between modelers and other stakeholders
  - —Through access on consistent AST data without SysML or tool knowledge
    - Enabling transition from document-based to model-based development
  - —Through central model-based data representation
    - Improving traceability by searching, comparing and cross-referencing of elements in MMS in the View Editor
  - —Adopted GitFlow[1] workflow for collaboration and reviews with branches and tags
  - —Implemented Digital Signoff mechanism
    - Supporting transition from static documents to live model-based views, by referencing model elements to be approved
  - —Implemented Issue Tracking with AST access, referencing model elements and comments in the View Editor

1) https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow

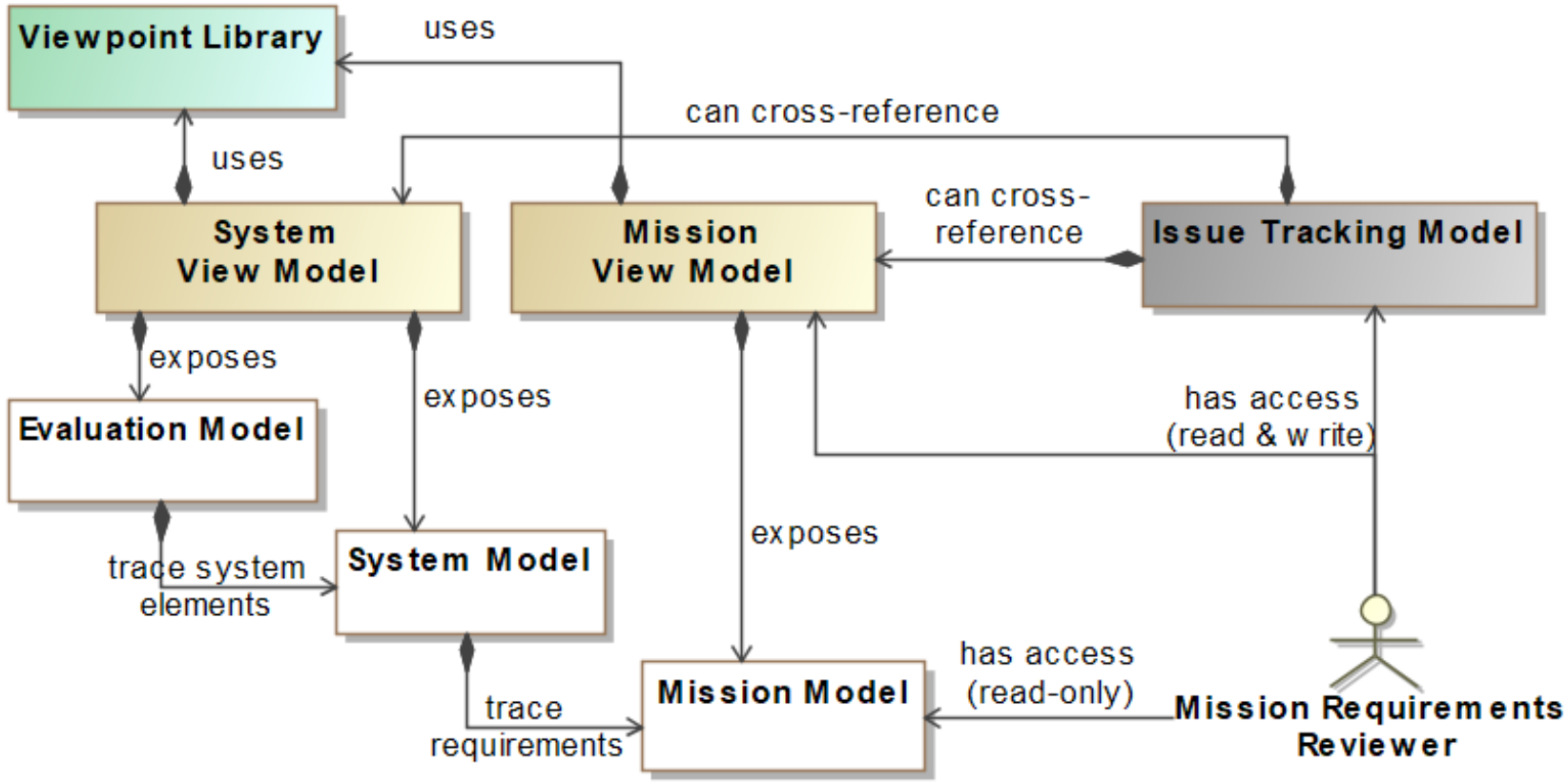
Guidelines and Findings (2)

- Use of DocGen & Viewpoint Library
  - —Model-based document generation using DocGen, enabling fast and useful design iterations
  - —Providing standardized viewpoints to quickly create consistent documents
    - Most viewpoint needs covered by the provided elements and OCL[1] constraints
    - Few modelers need to be familiar with DocGen
    - Allowing pre-planned view hierarchies to guide modeling by defining model structure and required content
- Required Modeling Considerations
  - —Consistent model structure to ensure viewpoints finding model elements
  - —Keeping document creation in mind while modeling
    - E.g. limiting diagram size and adding documentation
  - —Project usage mechanism for modularization, reuse, partitioning, traceability and user access

1) OCL = Object Constraint Language

Project Usage & User Permissions: Example

- Editing & commenting in the Mission View Model, without permission to directly change exposed requirements from Mission Model
- Creating issues in Issue Tracking Model, e.g. using prior comments



Composition ≙ Project Usage

Issues & Suggested Improvements

- Issues
  - —Errors preventing project usages in Teamwork Cloud
  - —Inconsistent document representation between SysML tool and View Editor
  - —View Editor creating names for elements unable to have a name in SysML (fixed)
- Suggested Improvements
  - —Utilizing open access to AST data, e.g. for ontological reasoning
  - —Augmented cross referencing in View Editor, e.g. for SysML term elements
  - —Improved creation of model elements in View Editor, e.g. using MapleMBSE
  - —Improved reviewing in View Editor, e.g. with a change request process involving tracked issues and signoffs that may prevent changes without an reversed approval status and notifications

Further Information

- OpenMBEE http://www.openmbee.org/
  - —GitHub: https://github.com/Open-MBEE
  - —OpenMBEE Public Server Information: https://github.com/Open-MBEE/open-mbee.github.io/wiki/OpenMBEE-public-server-information
- Surrogate Pilot
  - —Apan SET Surrogate Pilot Group: https://community.apan.org/wg/navair-set/set-surrogate-pilot/
  - —View Editor: http://ime.sercuarc.org/alfresco/mmsapp/mms.html (Login instructions available on Apan)

Thank you!

Dr. Benjamin Kruse

Research Assistant Professor

School of Systems & Enterprises

Systems Engineering Research Center

Stevens Institute of Technology