MARKLOGIC WORLD 19

Boeing: Querying model based systems engineering data

Brian Wirth, Java Developer, Boeing

Querying Model Based Systems Engineering Data

Brian Wirth
Rob Saunderson
Aleksander Przybylo

CONTENTS

- The why
  - The RAT and the Gimli Glider!
  - FAA requirements
- The what
  - SysML data
  - Network interface model
- The how
  - OpenMBEE environment
  - MarkLogic

The RAT and the Gimli Glider!

- https://en.wikipedia.org/wiki/Gimli_Glider
- Air Canada Flight 143 on July 23 1983: a Boeing 767 loses all power to both engines midflight
- Experts initially thought a faulty fuel gauge was to blame
- Real reason? Hint: it was the first metric plane in Canada
- Another hint? Specific gravity of jet fuel is 0.8 kg/L = 1.77 lbs/L
- The fuelers in Montreal only loaded half the necessary fuel

- The Ram Air Turbine automatically deployed
- Generated enough power to restore functions to flight control instruments, radio and some hydraulic systems
- The pilots successfully land the plane in Gimli, Manitoba
- 10 people sustained only minor injuries

FAA requirements

PART 25—AIRWORTHINESS STANDARDS: TRANSPORT CATEGORY AIRPLANES

Control Systems

§25.671 General.

(a) Each control and control system must operate with the ease, smoothness, and positiveness appropriate to its function.

(b) Each element of each flight control system must be designed, or distinctively and permanently marked, to minimize the probability of incorrect assembly that could result in the malfunctioning of the system.

(c) The airplane must be shown by analysis, tests, or both, to be capable of continued safe flight and landing after any of the following failures or jamming in the flight control system and surfaces (including trim, lift, drag, and feel systems), within the normal flight envelope, without requiring exceptional piloting skill or strength. Probable malfunctions must have only minor effects on control system operation and must be capable of being readily counteracted by the pilot.

(1) Any single failure, excluding jamming (for example, disconnection or failure of mechanical elements, or structural failure of hydraulic components, such as actuators, control spool housing, and valves).

(2) Any combination of failures not shown to be extremely improbable, excluding jamming (for example, dual electrical or hydraulic system failures, or any single failure in combination with any probable hydraulic or electrical failure).

(3) Any jam in a control position normally encountered during takeoff, climb, cruise, normal turns, descent, and landing unless the jam is shown to be extremely improbable, or can be alleviated. A runaway of a flight control to an adverse position and jam must be accounted for if such runaway and subsequent jamming is not extremely improbable.

(d) The airplane must be designed so that it is controllable if all engines fail. Compliance with this requirement may be shown by analysis where that method has been shown to be reliable.

[Doc. No. 5066, 29 FR 18291, Dec. 24, 1964, as amended by Amdt. 25-23, 35 FR 5674, Apr. 8, 1970]

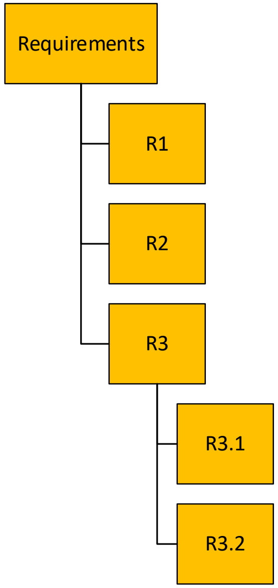
https://www.ecfr.gov/cgi-bin/text-idx?node=14:1.0.1.3.11

SysML data

- We use SysML to represent the requirements and their functional and logical product breakdowns (RFLP)
- SysML (Systems Modeling Language) is an extension of UML (Unified Modeling Language) for Systems modelling
- An OMG (Object Management Group) standard implemented by multiple MBSE software vendors

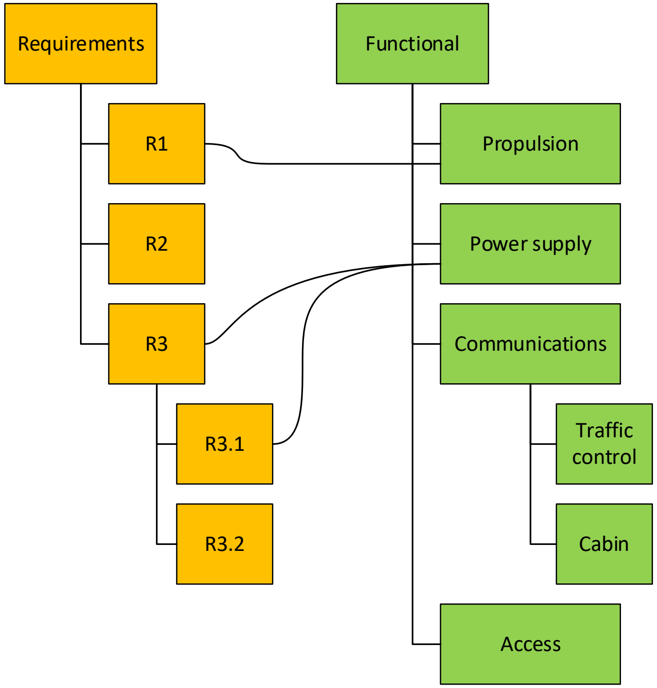
SysML data

- We use SysML to represent the requirements and their functional and logical product breakdowns
- FAA, Customer and Boeing requirements (~50K) are decomposed into lower tier requirements (~5M)



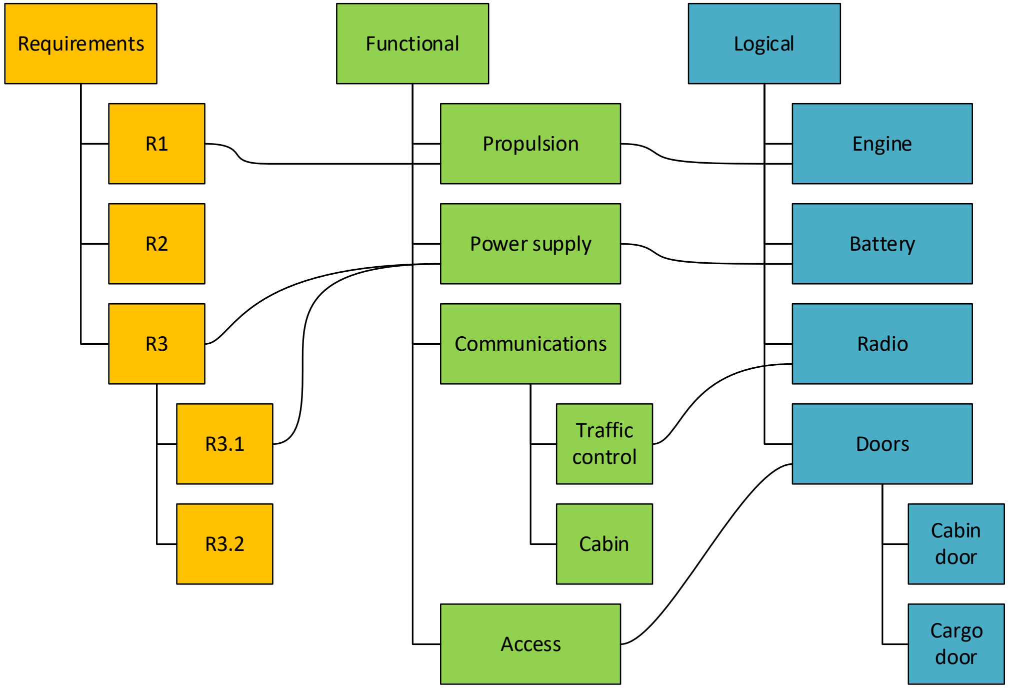
SysML data

- We use SysML to represent the requirements and their functional and logical product breakdowns
- FAA, Customer and Boeing requirements (~50K) are decomposed into lower tier requirements (~5M)
- Each requirement is allocated to a functional (propulsion, electrical power supply) and then logical (engine, battery) element in SysML – functional and logical architecture analysis and development validates lower tier requirements and their allocations



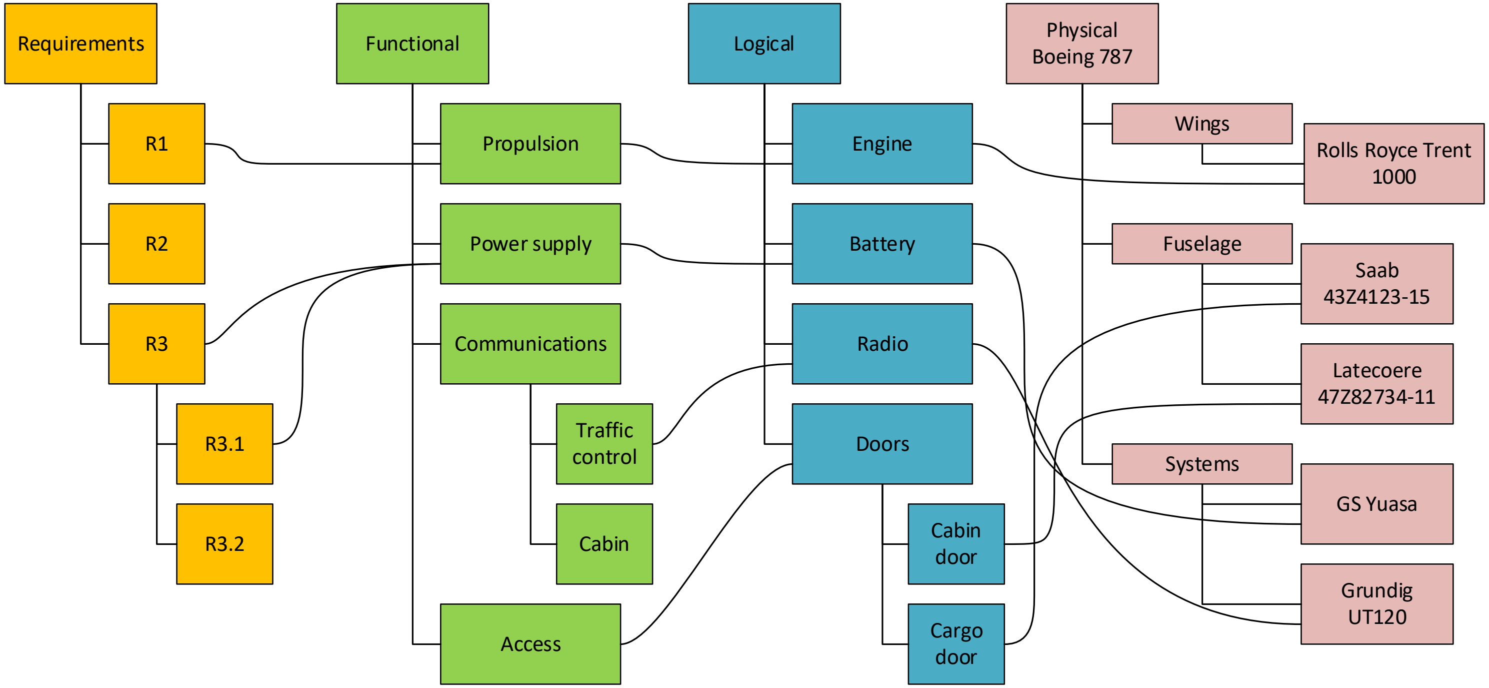
SysML data

- We use SysML to represent the requirements and their functional and logical product breakdowns
- FAA, Customer and Boeing requirements (~50K) are decomposed into lower tier requirements (~5M)
- Each requirement is allocated to a functional (propulsion, electrical power supply) and then logical (engine, battery) element in SysML – functional and logical architecture analysis and development validates lower tier requirements and their allocations



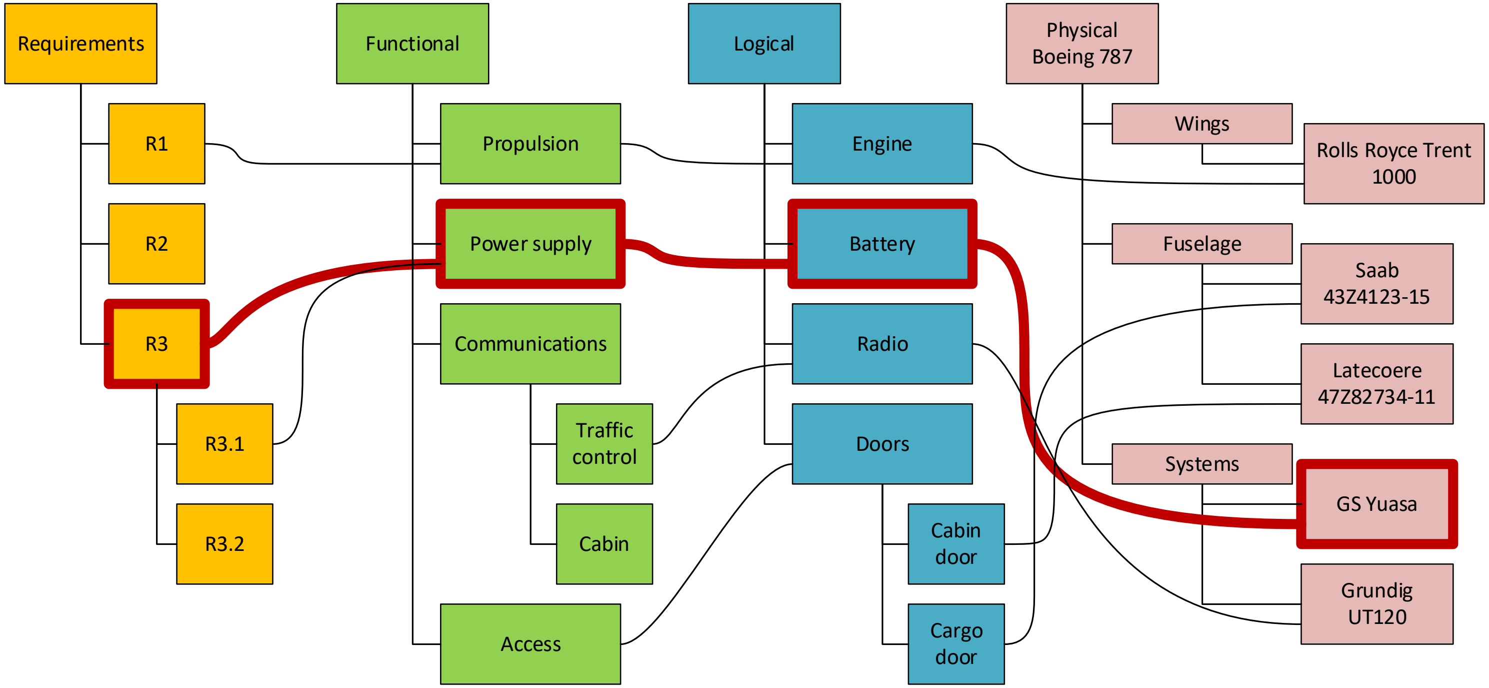
SysML data

- We use SysML to represent the requirements and their functional and logical product breakdowns
- FAA, Customer and Boeing requirements (~50K) are decomposed into lower tier requirements (~5M)
- Each requirement is allocated to a functional (propulsion, electrical power supply) and then logical (engine, battery) element in SysML – functional and logical architecture analysis and development validates lower tier requirements and their allocations
- Ultimately logical elements are implemented with a physical (RR Trent 1000) breakdown
  - The thousands of customer options, quickly evolving engineering, massive amount of parts (several million) and high production rate (2 airplanes per day on the 737 line) make the number of possible configurations practically infinite



SysML data

- We use SysML to represent the requirements and their functional and logical product breakdowns
- FAA, Customer and Boeing requirements (~50K) are decomposed into lower tier requirements (~5M)
- Each requirement is allocated to a functional (propulsion, electrical power supply) and then logical (engine, battery) element in SysML – functional and logical architecture analysis and development validates lower tier requirements and their allocations
- Ultimately logical elements are implemented with a physical (RR Trent 1000) breakdown
  - The thousands of customer options, quickly evolving engineering, massive amount of parts (several million) and high production rate (2 airplanes per day on the 737 line) make the number of possible configurations practically infinite



Querying and analytics

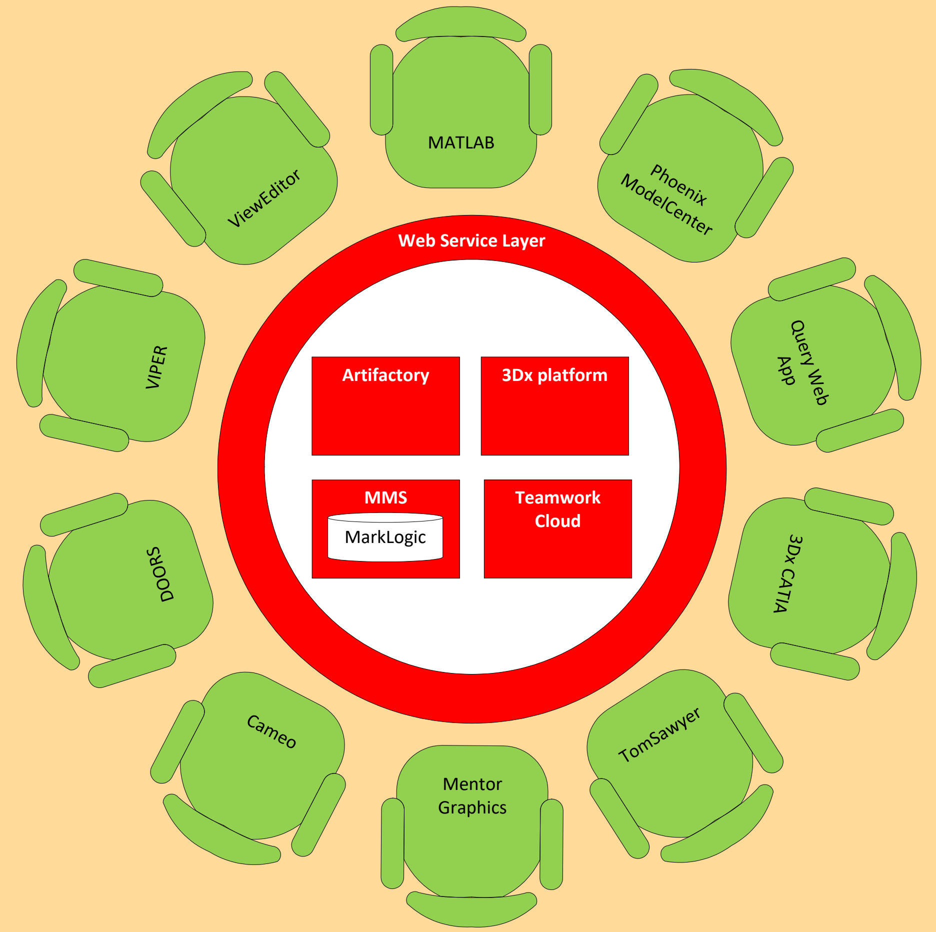
- The SysML data represents the requirements and their functional and logical product breakdowns – millions of objects, relations, and associated interdependencies
  - Analysis of all the combinations is increasingly difficult
- High performance querying and analytics capability is needed to validate the propagation of each requirement across all the intermediary structures
  - Perform impact analysis (what if…)
  - Perform validation (all requirements are satisfied for all implemented configurations)
  - Simulate the model: **given the failure of both engines, does the RAT deploy and supply power to all critical components?**
- Strategy
  - SysML data modeled in No Magic Cameo is stored in MarkLogic and converted to an RDF graph
  - The entire network model (integrating all airplane subsystems) is also converted to an RDF graph stored in MarkLogic
    - Includes all modules, ports and all possible signals down to the bit level
    - Results in several million triples for a single airplane configuration (much more to cover all configurations)

OpenMBEE

OpenMBEE = Open Model Based Engineering Environment (http://www.openmbee.org/)

- Open source community project aimed at enabling team collaboration and integration in large scale projects
- Initially developed by JPL
- Actively used by Lockheed Martin, Boeing, Ford and others, mainly as MBSE platform

Open MBEE reference architecture ("Seat at the table")



Adapted from: http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:incose_mbse_iw_2019:casestudy_europalanderconcept.pdf

Why MarkLogic?

- Support of W3C standards: RDF, SPARQL
- Built-in configuration management of triples
  - We only manage at the document level
  - The mapping of triples to documents is preserved and we do not need to manually identify triples affected by a document change
- Scalability and performance
  - Accumulation of historical data does not affect query performance
- Additional out-of-the-box features
  - Temporal queries
  - Collection management
  - Multi-model DB

Questions?