33rd Annual INCOSE international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023

Nils Kuelper, Thimo Bielsky, Jasmin Broehan, Frank Thielecke

Hamburg University of Technology, Institute of Aircraft Systems Engineering, Hamburg, Germany

Model-based Framework for Knowledge-Driven Systems Architecting Demonstrated on a Hydrogen-Powered Concept Aircraft

AFST

TUHH

Need for Model-Based Framework

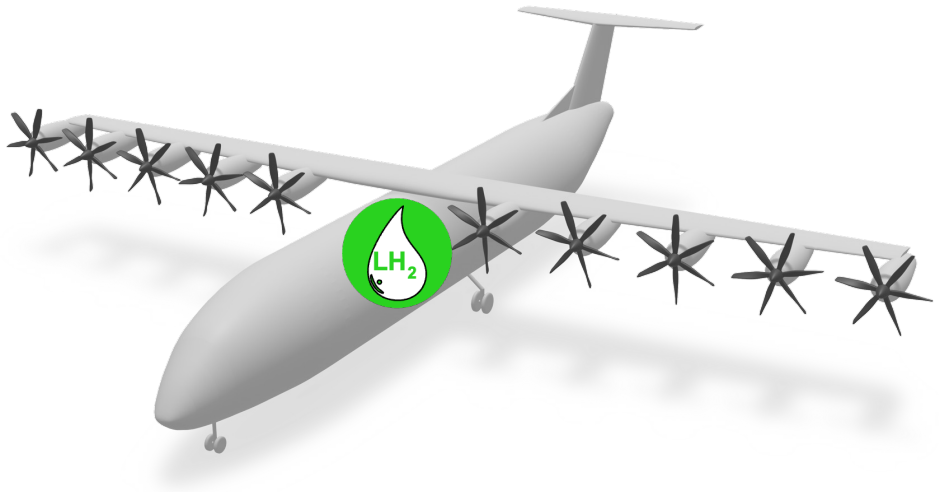
Goal: Sustainable aviation → **Hydrogen**

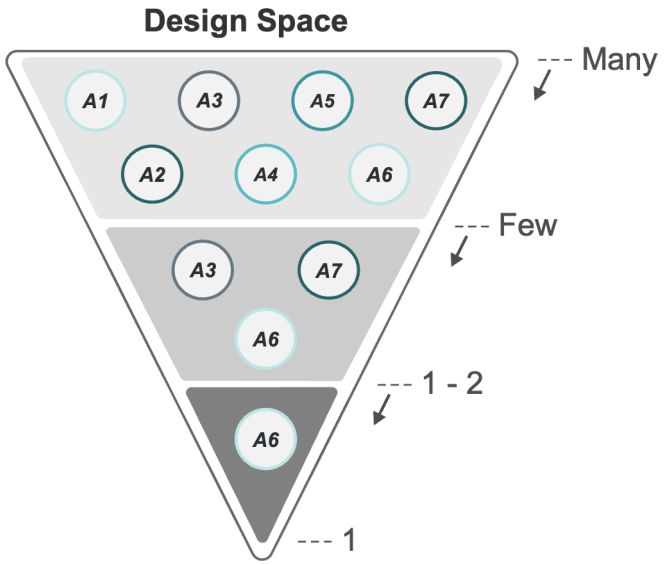
Disruptive technology concepts → **Uncertainty**

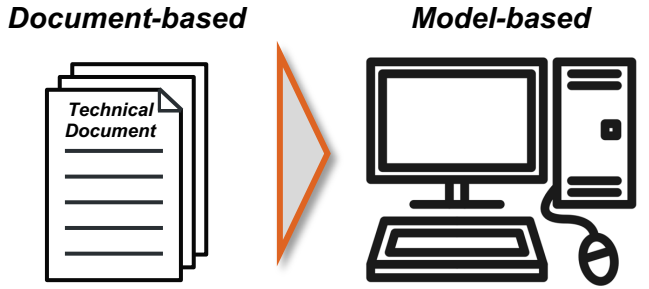
Vast design space with many interdependences → **Complexity**

Consider well understood aircraft → **Knowledge basis**

Conserve insights of trade studies → **Knowledge gain**







Improved model-based framework for H2 powered aircraft necessary to handle uncertainty and complexity accelerating aircraft development process.

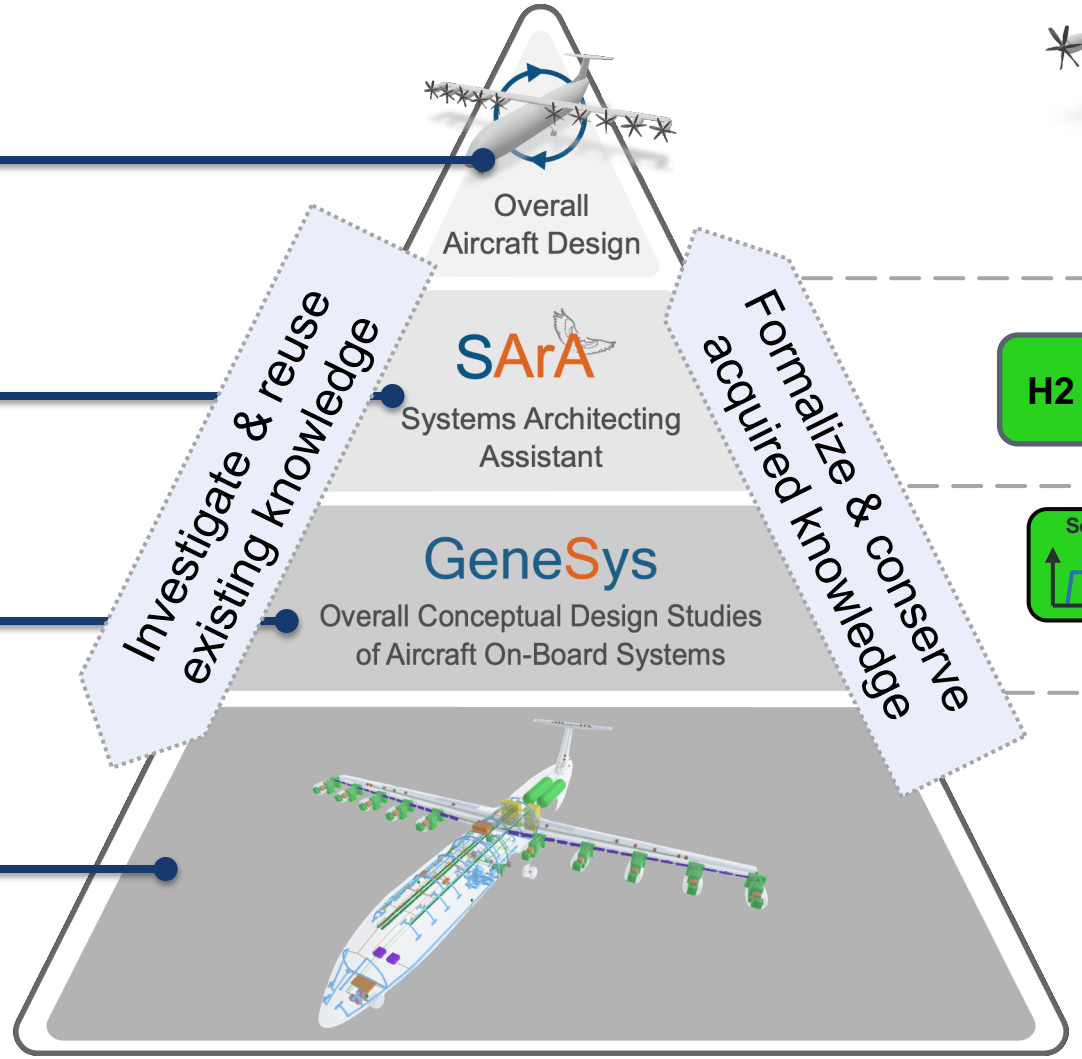
Aircraft Conceptual Design Process

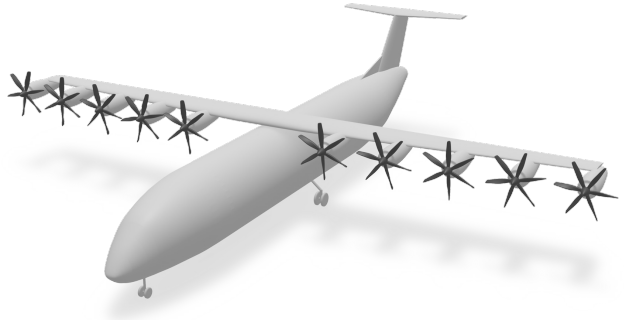
Aircraft geometry and Top-Level Aircraft Requirements (TLARs)

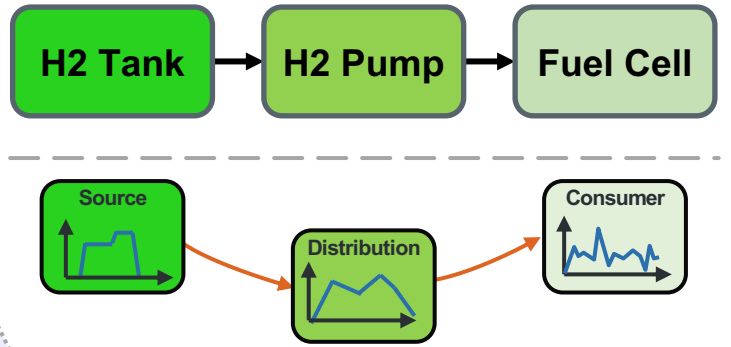
Functional-logical architecture definition & design space reduction

Physical component sizing based on sizing laws for technology trade studies

Preliminary feasible hydrogen-powered concept aircraft with OBS







Conserving and reusing knowledge to accelerate future aircraft development.

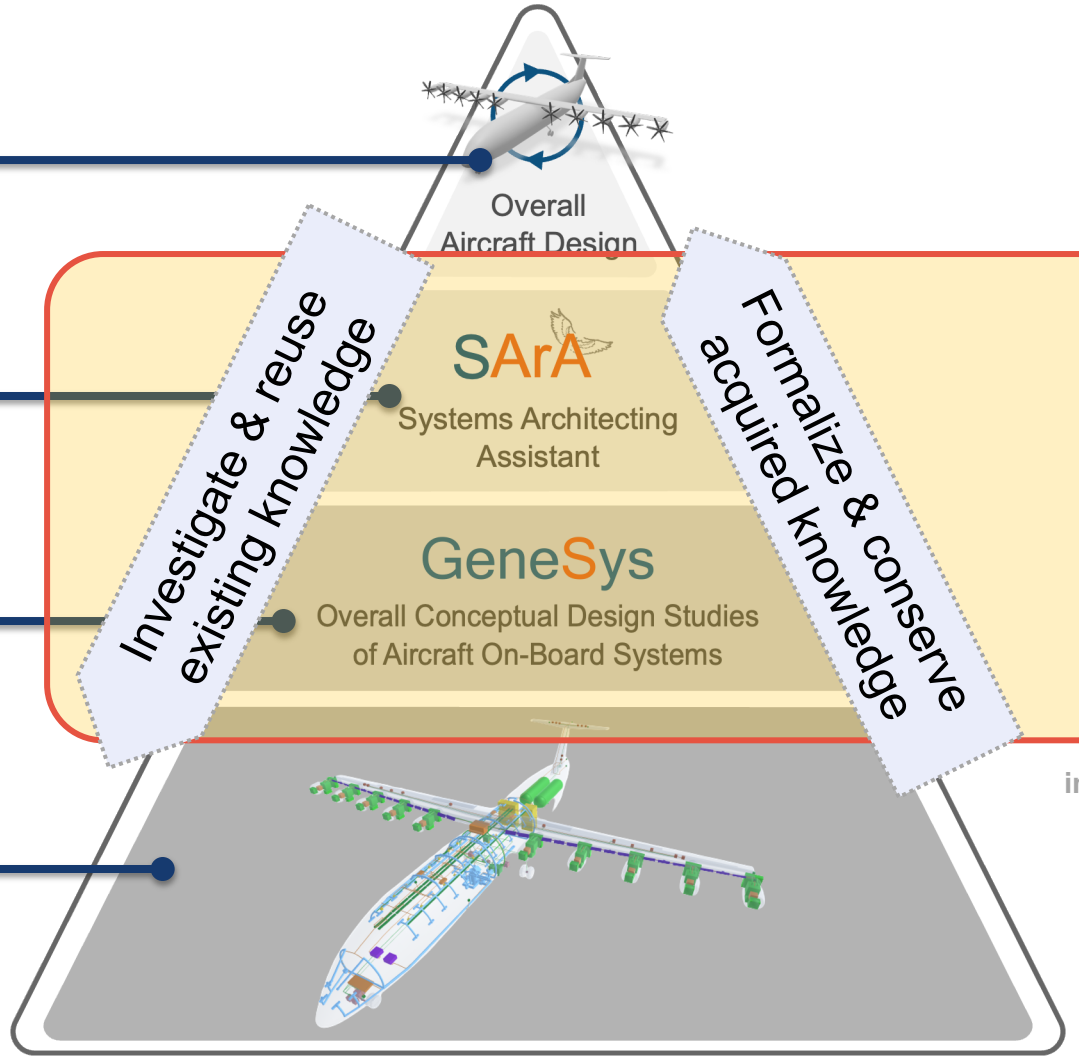
Knowledge Management Methodology - *iDAM*

Aircraft geometry and Top-Level Aircraft Requirements (TLARs)

Functional-logical architecture definition & design space reduction

Physical component sizing based on sizing laws for technology trade studies

Preliminary feasible hydrogen-powered concept aircraft with OBS

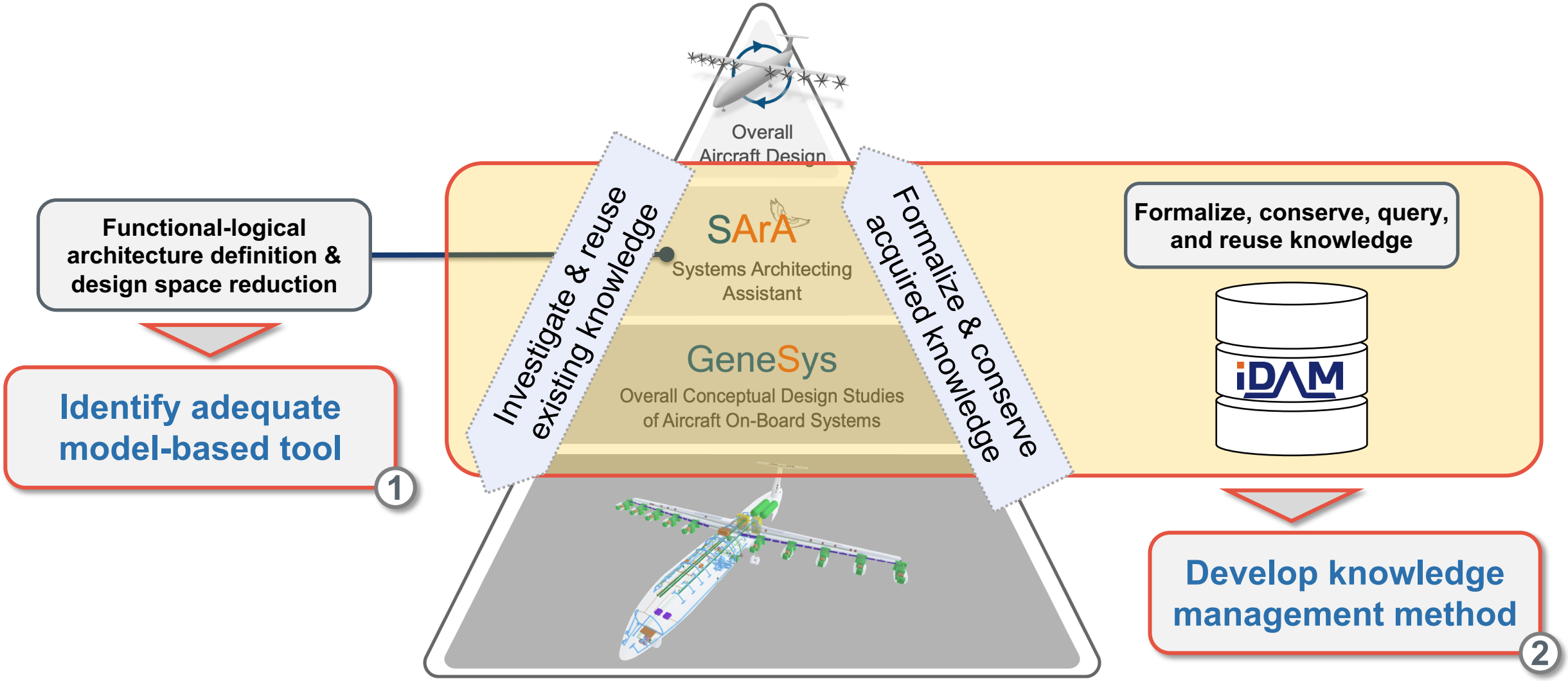


Formalize, conserve, query, and reuse knowledge

intelligent Data Analytics and Management

***iDAM* is used to conserve, investigate and reuse systems architecture knowledge.**

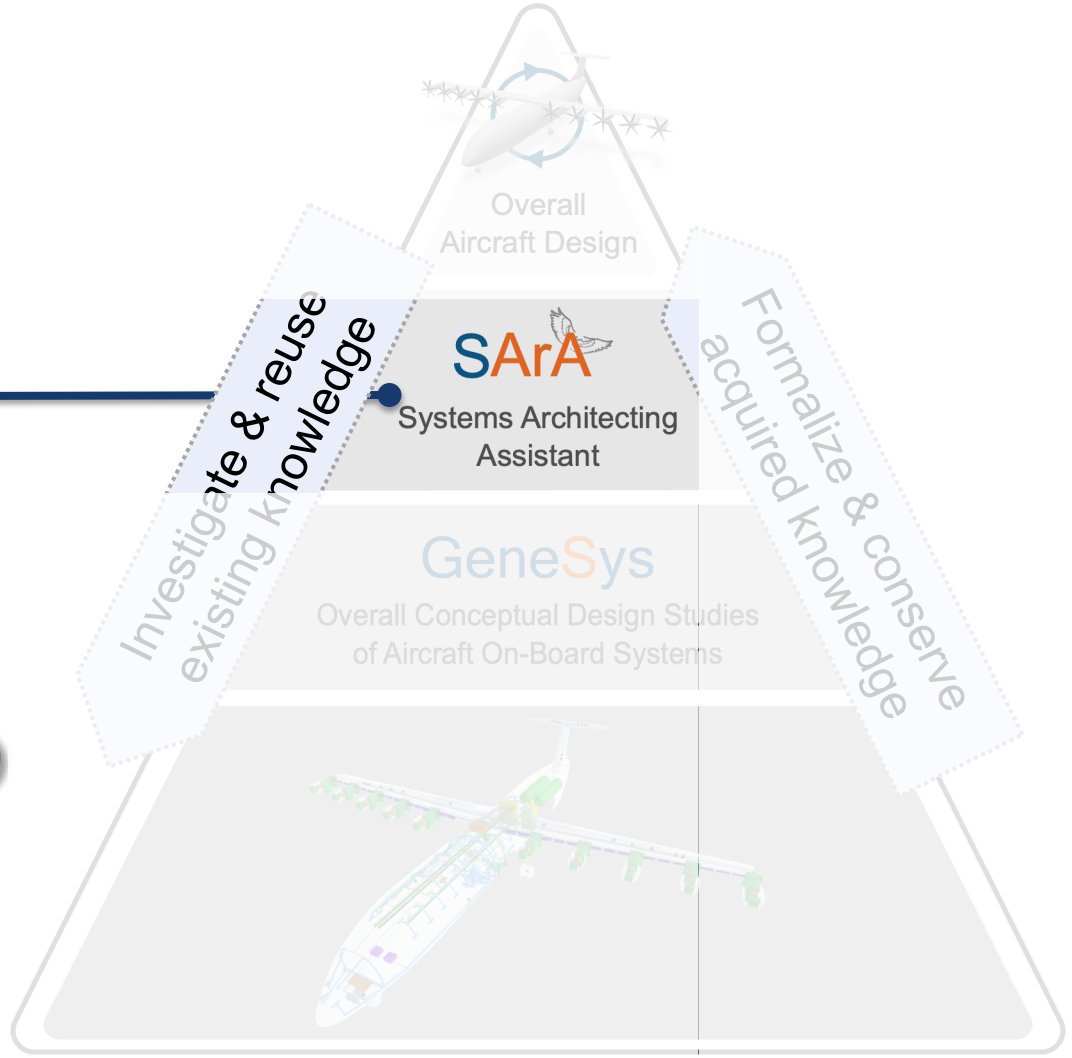
Missing Aspects for Aircraft Conceptual Design



Missing Aspects for Aircraft Conceptual Design

Functional-logical architecture definition & design space reduction

Identify adequate model-based tool (1)



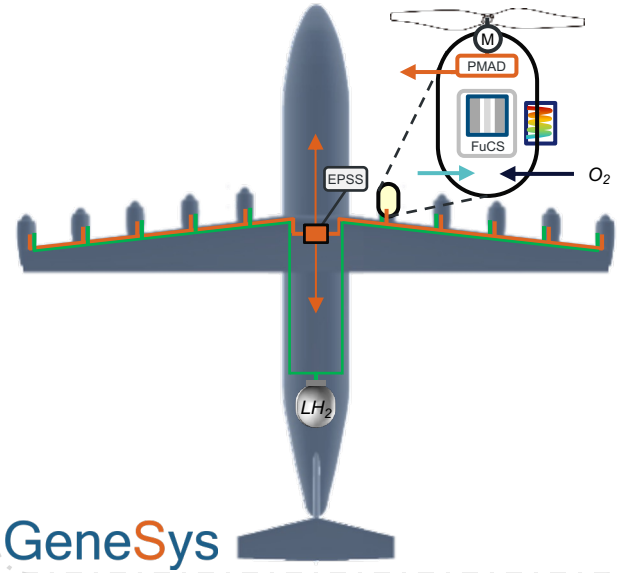
Formalize, conserve, query, and reuse knowledge

Develop knowledge management method (2)

Knowledge-Driven Systems Architecting Process with *SArA*

Overall Aircraft Design Level

Overall Systems Design Level



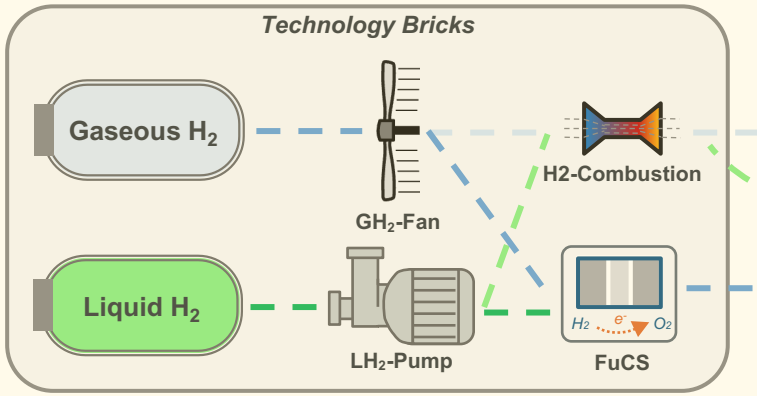
GeneSys

Systems Architecting

SArA

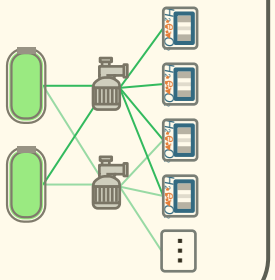
Tool needed

Technology Bricks



A1
A3
A2

Defined Architectures



Architecture Pool

A1 A3 A4 A2 A5

Validation (System Requirements)

Validation (Safety & Reliability)

Evaluation (Performance)

A2 A4

Model-based Tools for Systems Architecting

Tool evaluation based on well known and understood *Airbus A320* FCS architecture.

Evaluation Criteria for MBSE Tools of Systems Architecting

Criteria Prioritization and Weighting

- “Must-have” criteria (3)
- “Should-have” criteria (2)
- “Could-have” criteria (1)

Evaluation Approach

- Quality function deployment method
- Zero to five points scoring
  - Zero: not implemented
  - One: poor fulfillment
  - Five: complete satisfaction

$$S = \sum_{i=1}^{n} w_i \cdot r_i$$

S: Total score
w_i: Weighting factor
r_i: Rated points

Tool capabilities

- Analysis Capability
- Graphical Interface
- Model Modifiability
- Automation Capabilities
- Concurrent Modeling
- Constraint Checker
- Future Prospect
- Model Reusability
- Model Scalability

Modeling language

- Abstraction Capability
- Complexity Handling
- Traceability
- Modeling Standardization
- Variants Handling

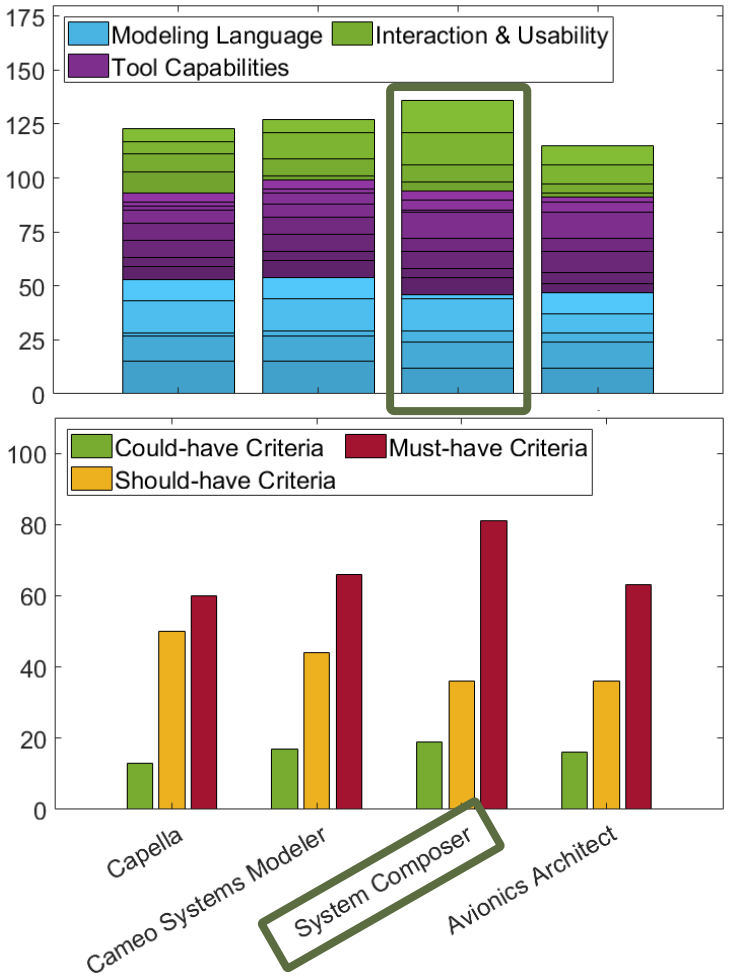
Interaction and Usability

- Interaction with *GeneSys*
- Interaction with *iDAM*
- Multidisciplinary Modeling
- User guidance

Results of MBSE Tool Evaluation Including Selection

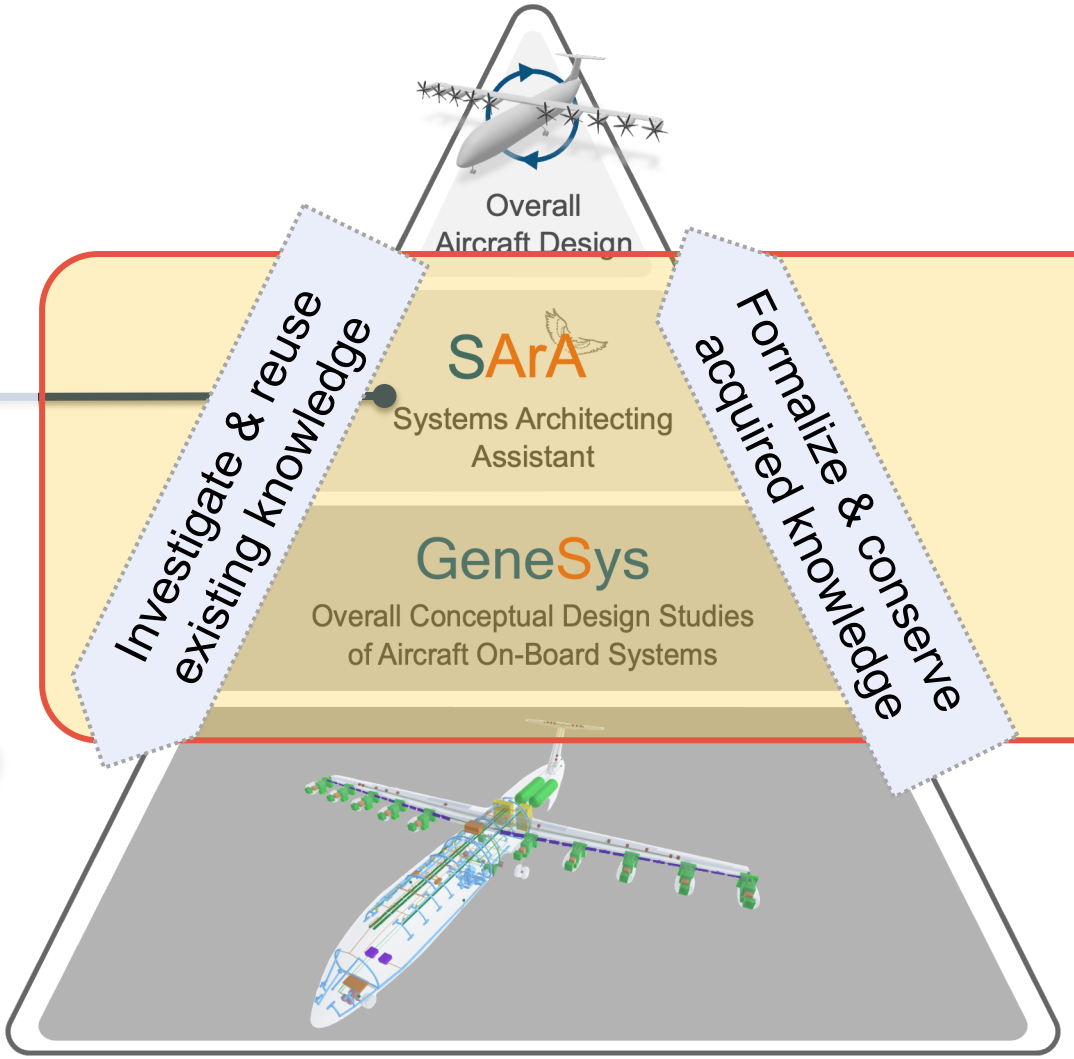
Result Characteristics

- Overall similar results
  - ➢ all suitable tools for knowledge-based systems architecting
- *System Composer* most beneficial for
  - − Interaction and usability
- *Capella* and *Cameo Systems Modeler* most beneficial for
  - − Modeling language
- *System Composer* scores highest in “must-have” criteria



System Composer **identified as most suitable tool for model-based, knowledge-driven systems architecting for presented application case.**

Missing Aspects for Aircraft Conceptual Design



Functional-logical architecture definition & design space reduction

Identify adequate model-based tool

1

Formalize, conserve, query, and reuse knowledge

iDAM

Develop knowledge management method

2

Method for Conserving Insights and Knowledge

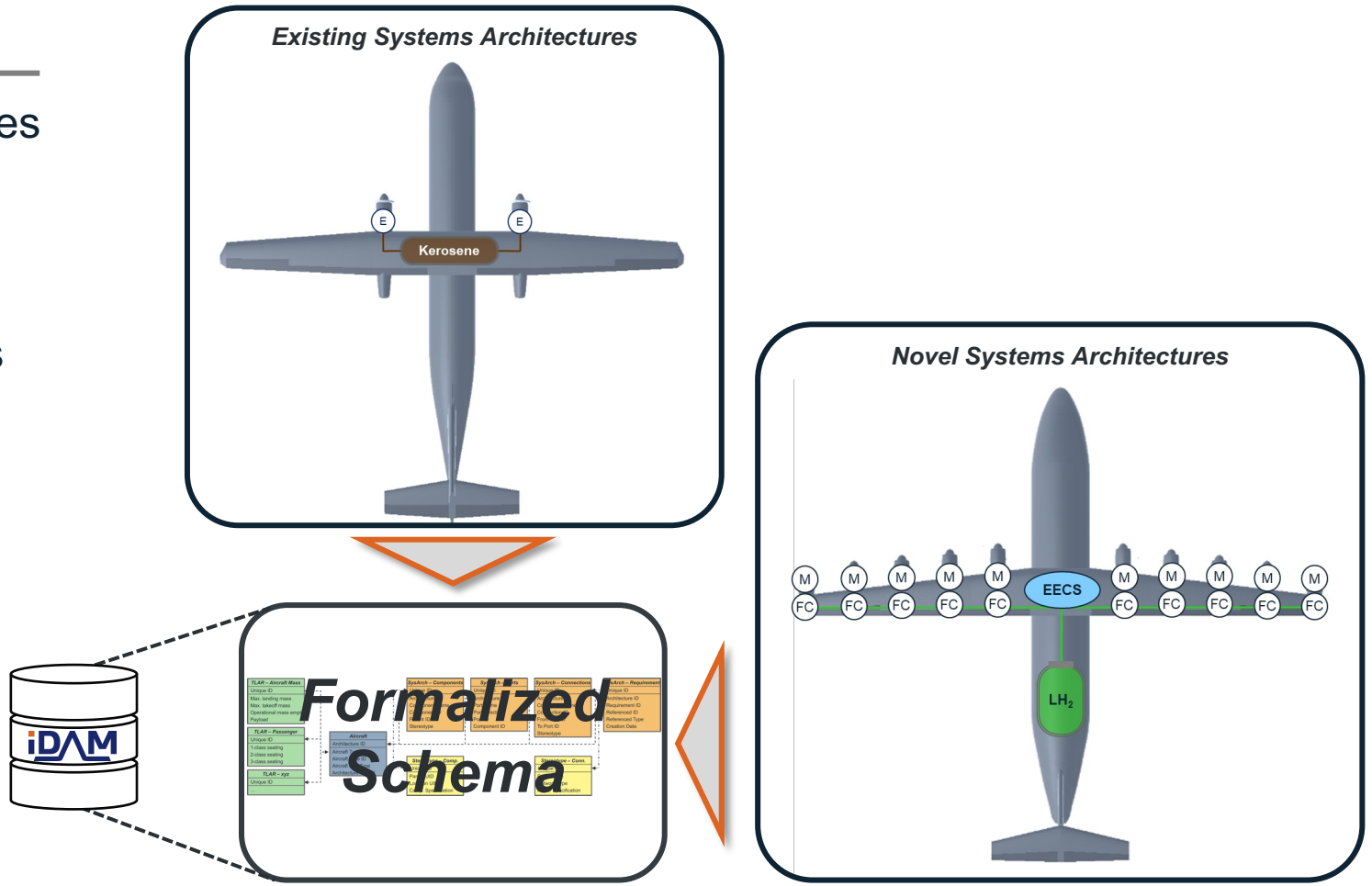
Requirements for Storing Method

- Existing on-board systems architectures
- Novel on-board systems architectures
- Handling of big data

- Independent of used architecting tools

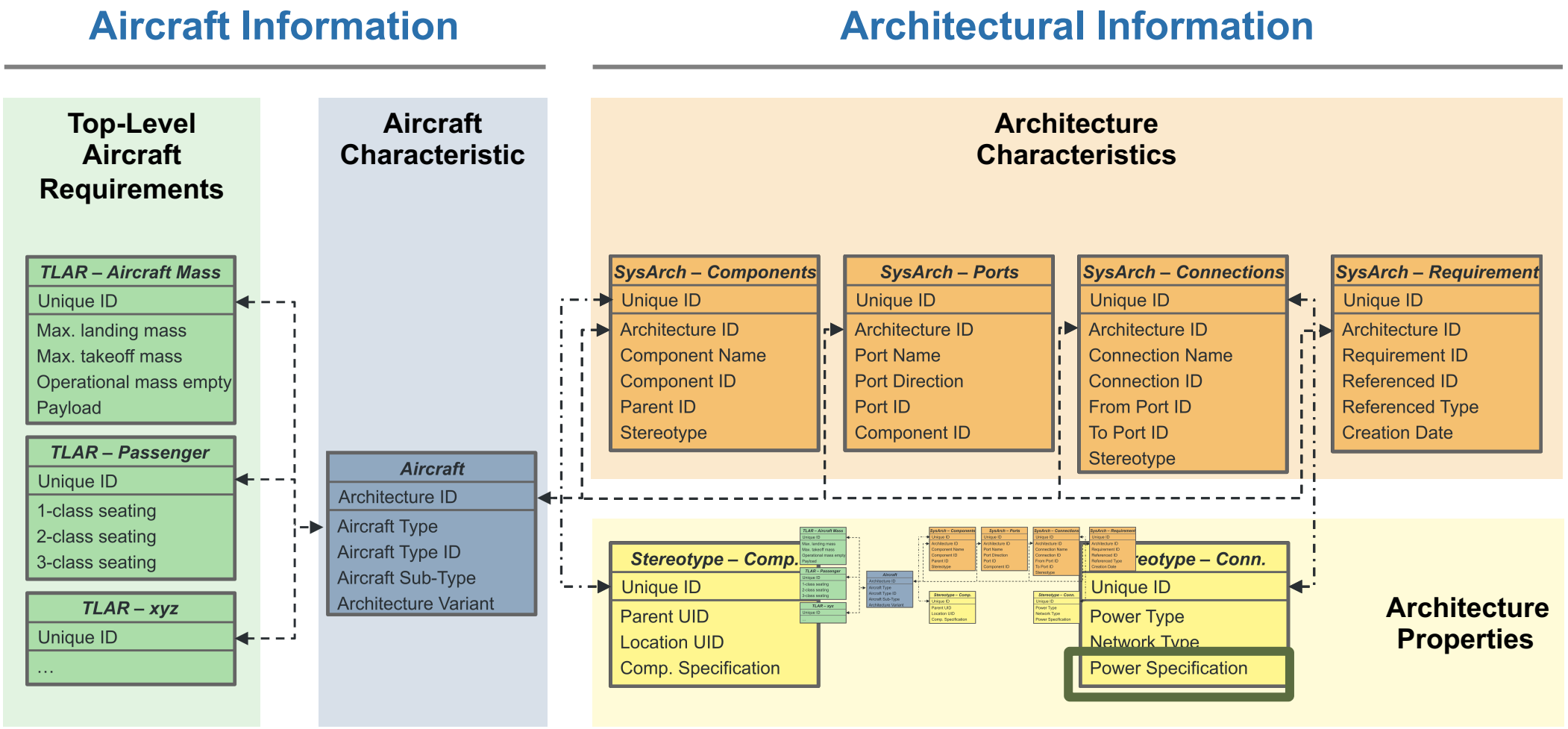
- Categorizable
- Linkable
- Queryable
- Reusable

- Applicable for *iDAM*
  - Relational database



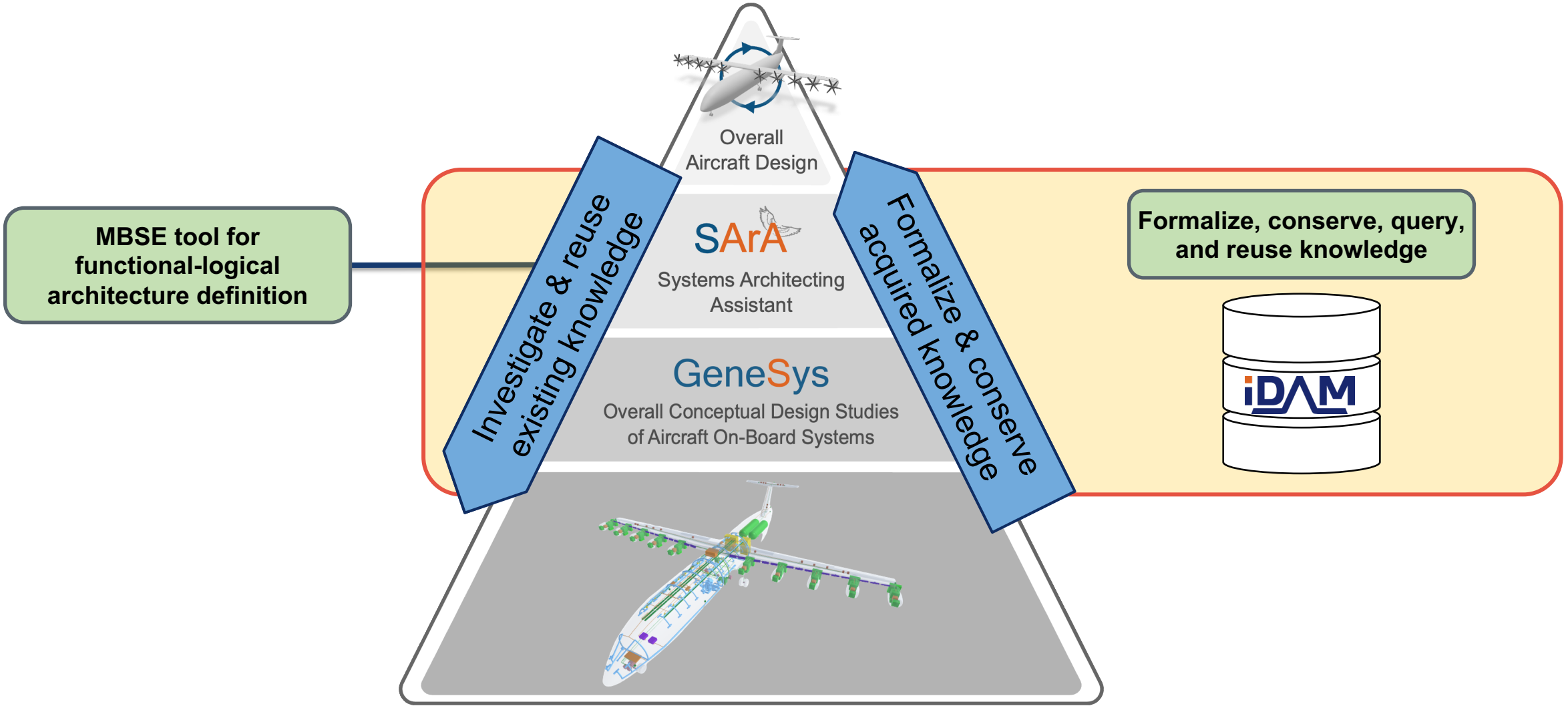
Knowledge about systems architectures is stored in relational database.

Formalized Schema for Knowledge Conservation

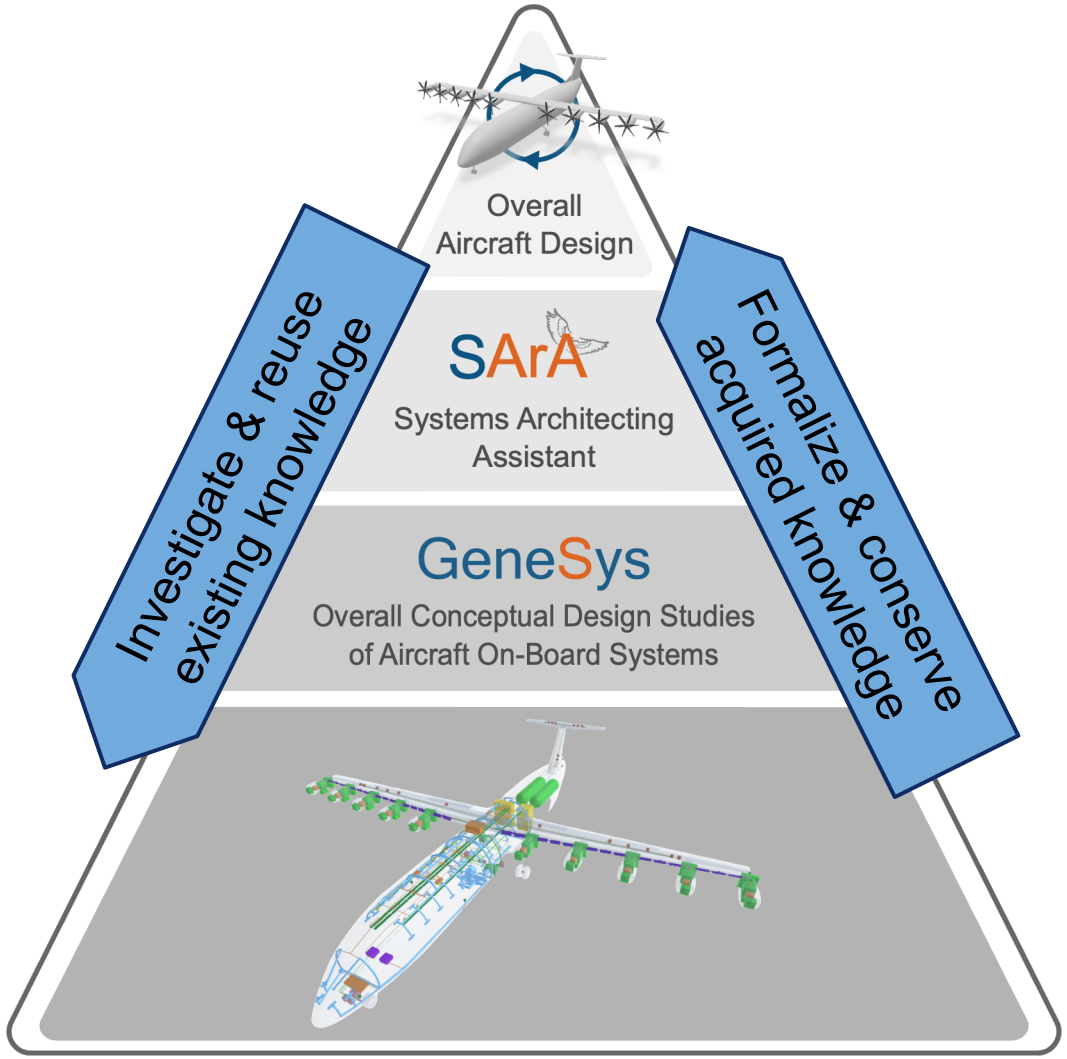


Database schema to formalize, link, conserve, query, and reuse knowledge.

Conceptual Design Demonstrated on Hydrogen Aircraft



Conceptual Design Demonstrated on Hydrogen Aircraft

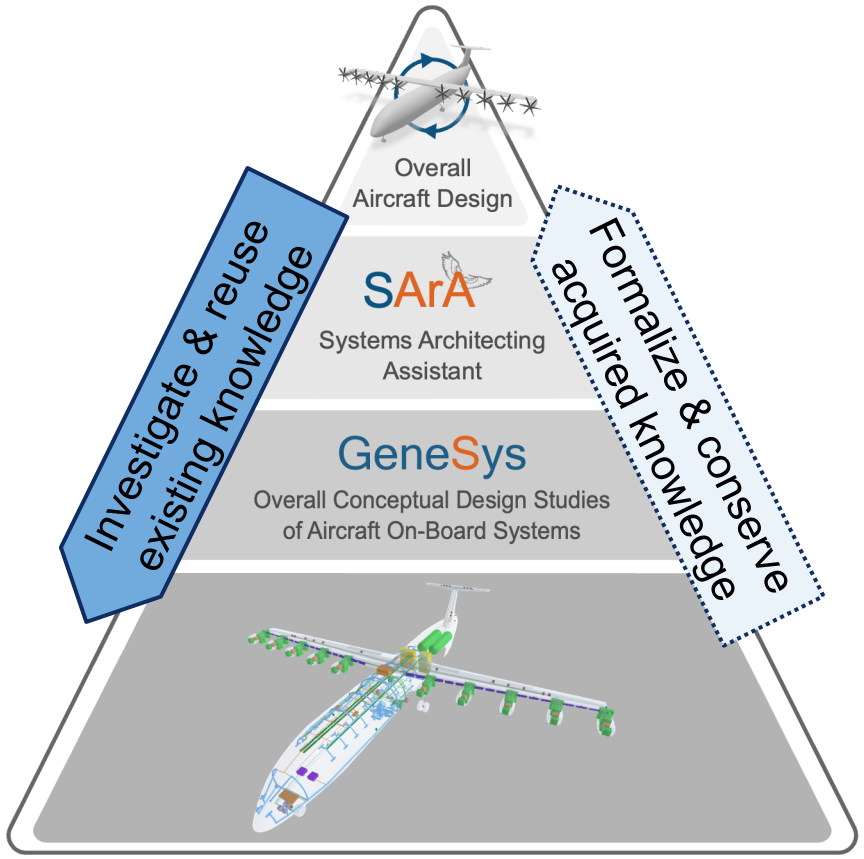


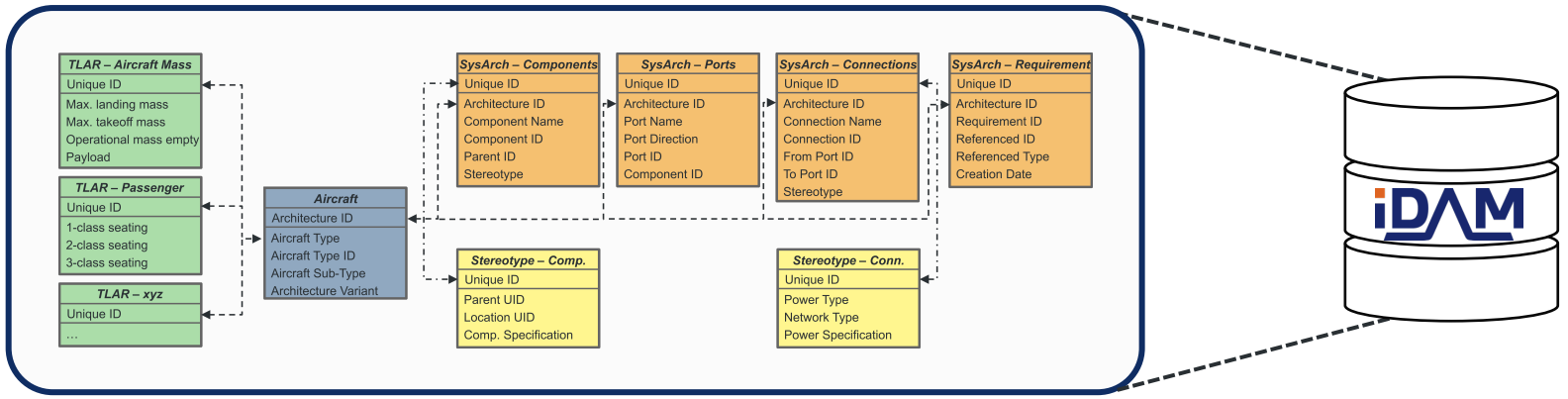
Concept Aircraft Characteristics

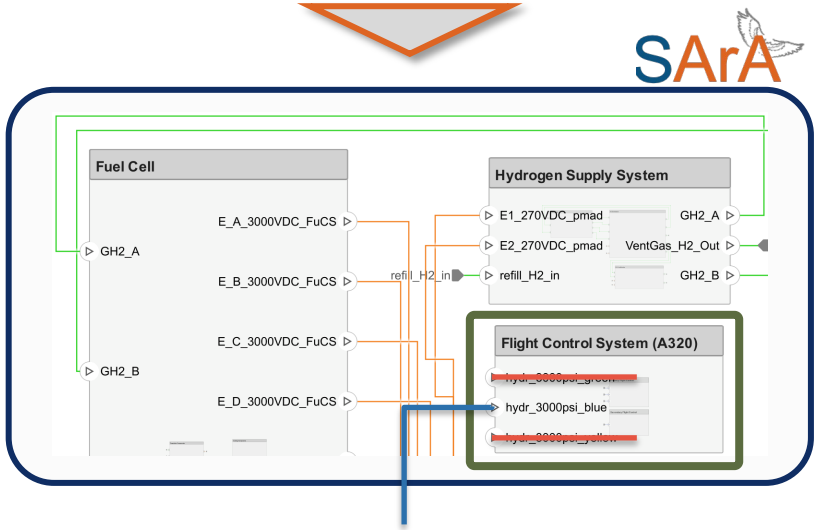
- Based on *ATR 72*-like aircraft
- Seating capacity of 70 passengers
- Liquid hydrogen power source
- 10 stand alone engine units (Pods)

- More-Electric-Aircraft on-board systems architecture
  - No more bleed air due to fuel cells
  - Electrified systems such as hydraulic power supply

Reuse of Architecture Knowledge for Revolutionary Design







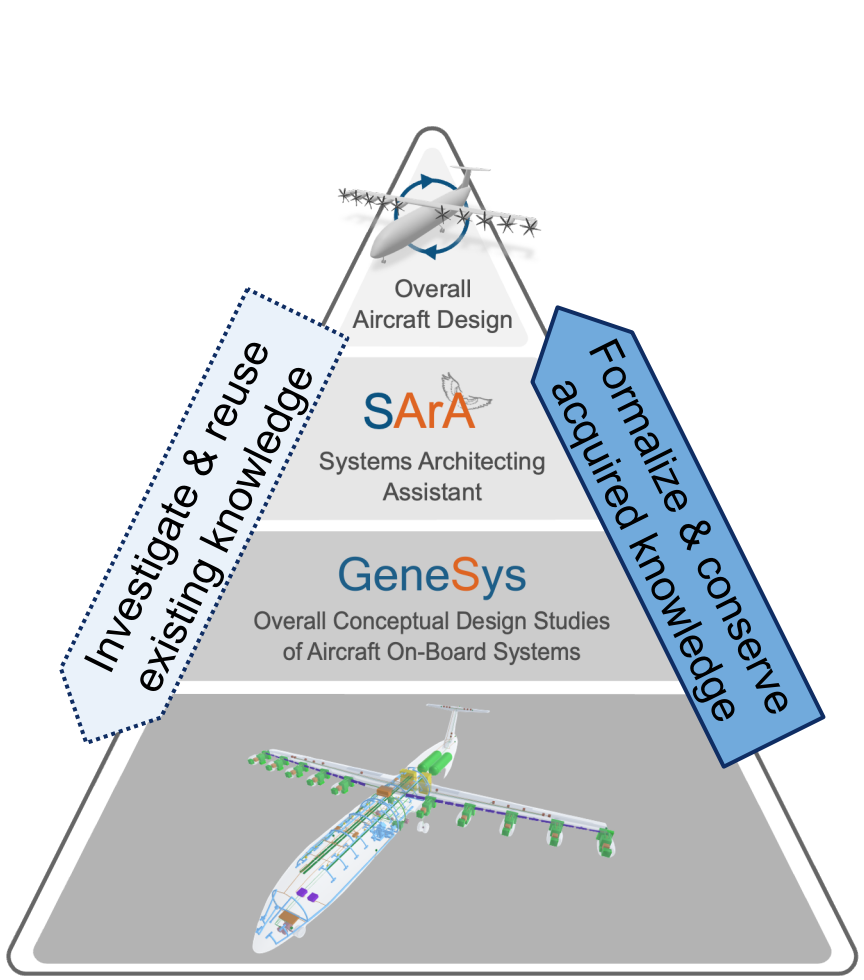
Safety Criticality

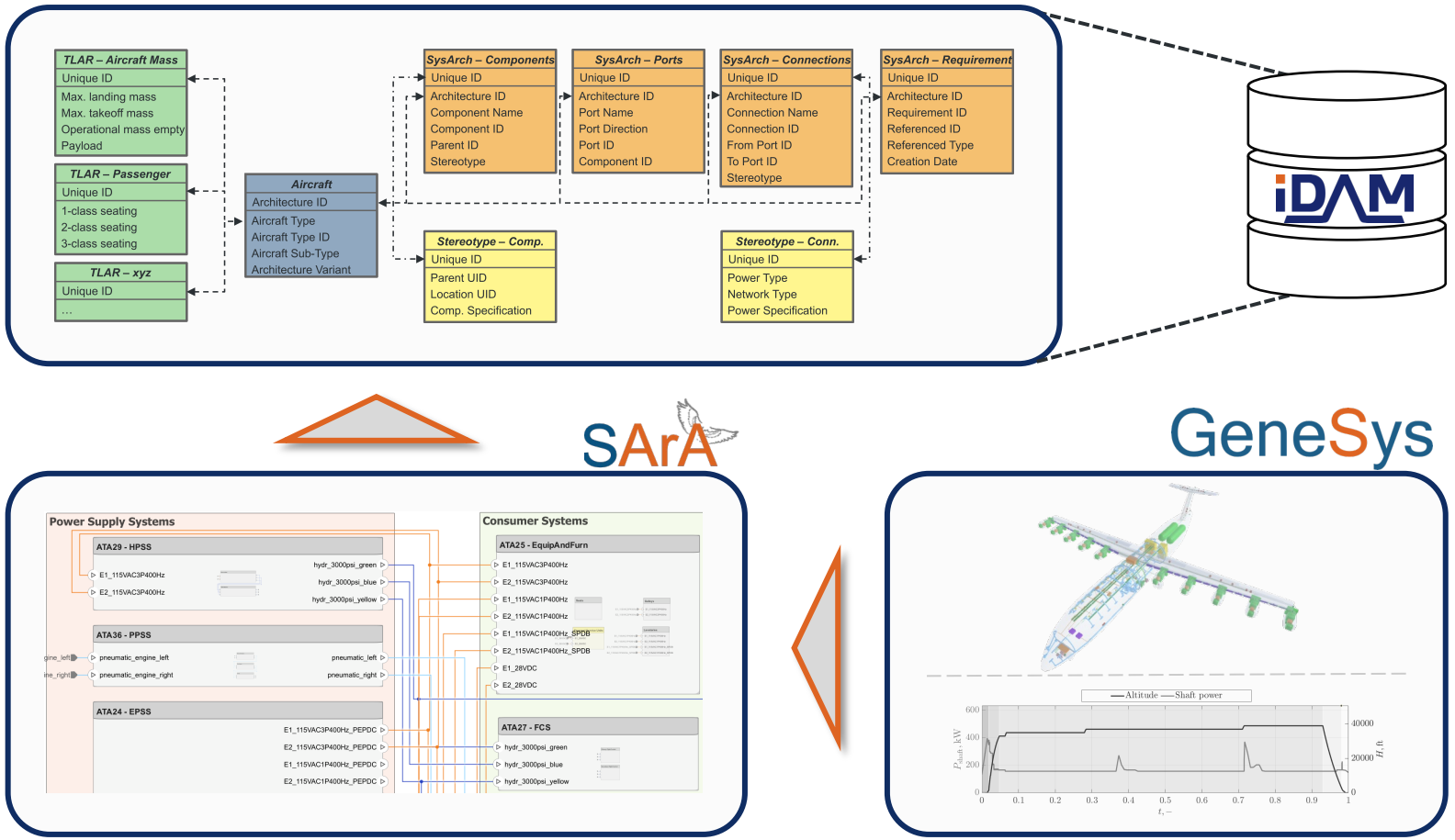
- Fulfillment of safety already during conceptual design
- FCS highly safety critical
  - Unknown solution for this aircraft

➢ Automated reuse of knowledge base on *iDAM*

Investigate and reuse stored knowledge as starting point for architecting process.

Conservation of Gained Architecture Knowledge

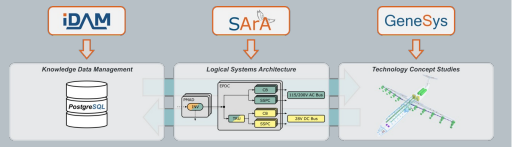




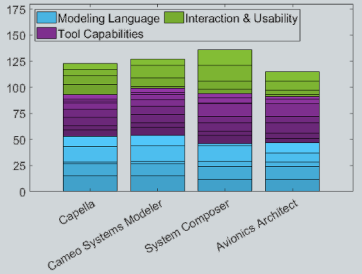
Conservation of newly acquired insights and knowledge about systems architectures.

Conclusion

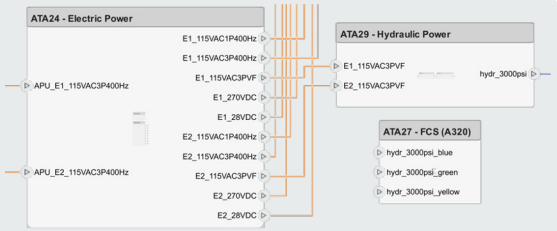
- Model-based framework for architecting
  - Systems architectures stored in database
  - Queryable and reusable formalized knowledge
  - Architecting process accelerated



- MBSE tool evaluation and selection
  - Identification of most suitable architecting tool
  - Capella, Cameo, Avionics Arch., System Composer
  - MathWorks System Composer selected



- Application to Hydrogen Concept Aircraft
  - Manual task remaining
  - Automated generation of novel architectures
  - Conservation of new insights



33rd Annual INCOSE international symposium

hybrid event

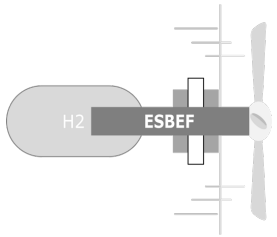
Honolulu, HI, USA
July 15 - 20, 2023

Thank you for your attention!

Supported by:

Federal Ministry for Economic Affairs and Climate Action

on the basis of a decision by the German Bundestag



TUHH

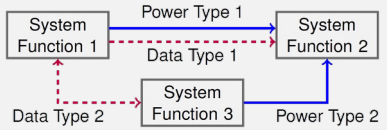
Contact
Nils Kuelper, M.Sc.
Hamburg University of Technology (TUHH)
Institute of Aircraft Systems Engineering (FST)
nils.kuelper@tuhh.de

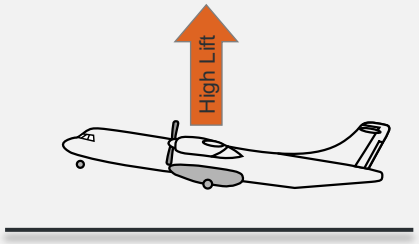
Backup

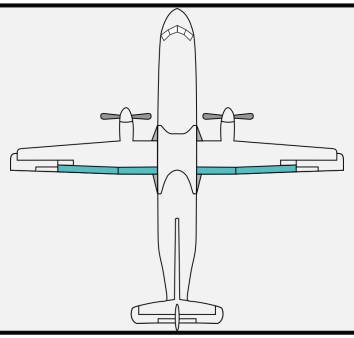
Nomenclature – Functional, Logical, Physical Architecture

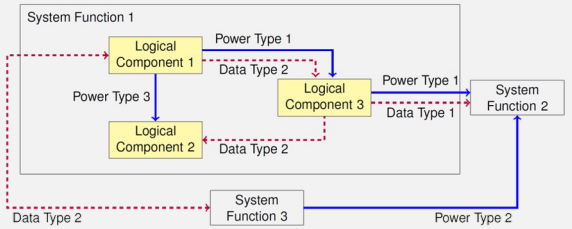
Functional Architecture

- Definition
  - Decomposition of system and subsystem functions, and their interdependencies
- Example: *Generate high lift*







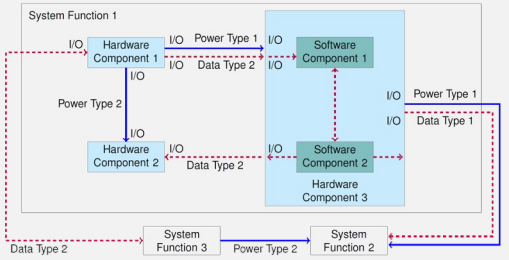


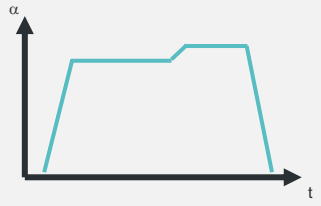
Logical Architecture

- Definition
  - High level component blocks and their data and power exchange fulfilling intended system functions
  - Only pre-selection of technology concepts
- Example: *Fowler flaps*

Physical Architecture

- Definition
  - Physical hardware components and their interrelations
  - Detailed technology selection and system behavior
- Example: *Geometry and behavior of fowler flap*



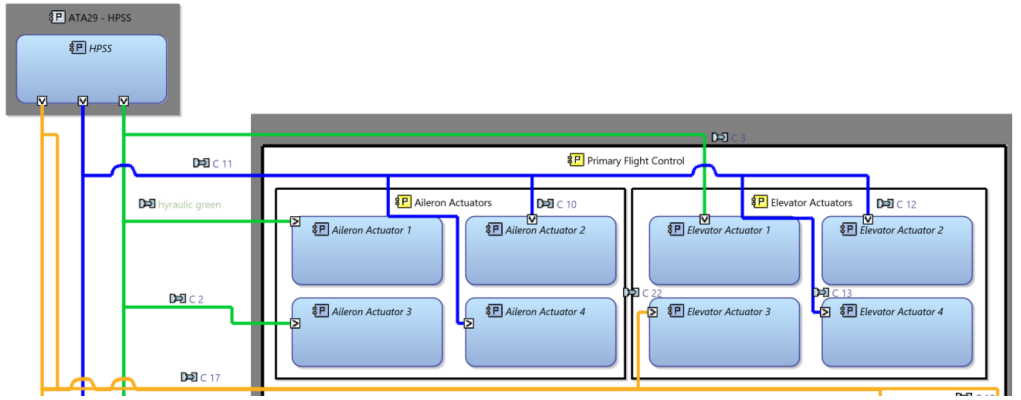


Evaluation of Capella and Cameo Systems Modeler

Capella

- ✓Different levels of abstraction
- ✓Standardized modeling language (*SysML*)
- ✓User guidance (*ARCADIA*)

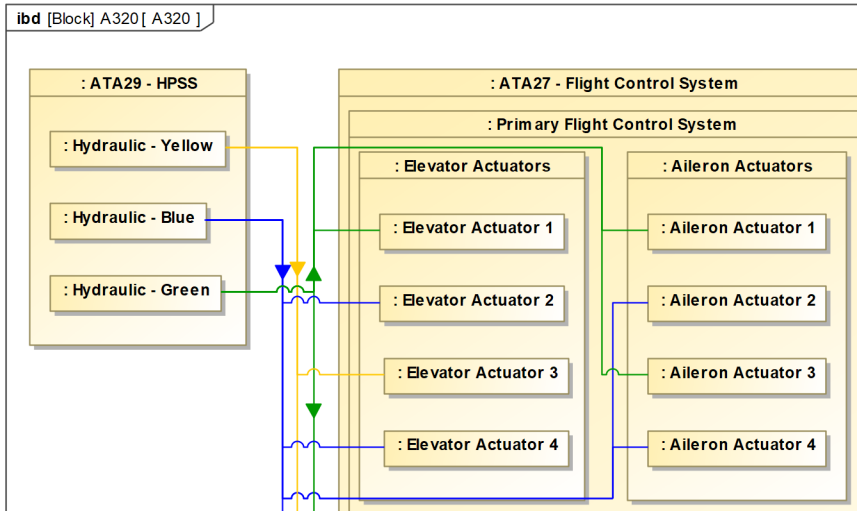
- – Direct connection per default
- – Exporting architecture to GeneSys not trivial



Cameo Systems Modeler

- ✓Abstraction capabilities
- ✓Standardized modeling language (*SysML*)
- ✓Behavior diagrams

- – No direct variant handling
- – Missing user guidance for architecting

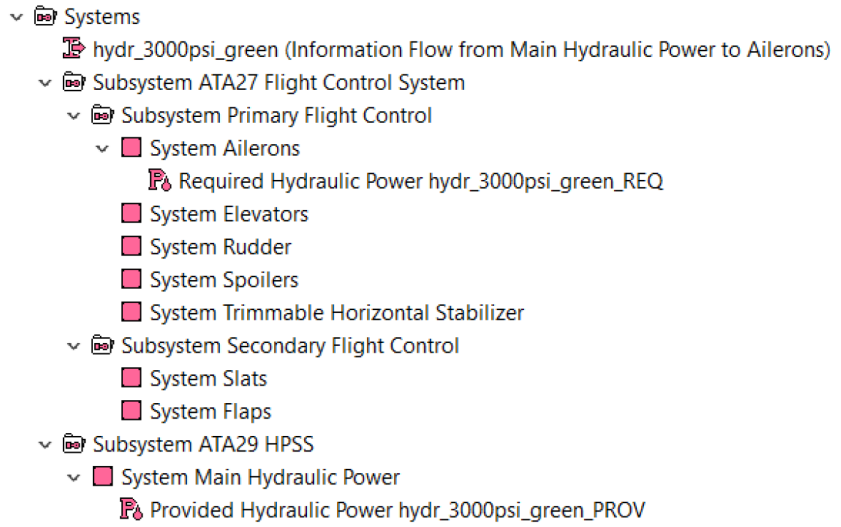


Evaluation of Avionics Architect and System Composer

Avionics Architect

- ✓Tree-based, hierarchical structure
- ✓Import, export, optimize architectures
- ✓Standardized modeling language (meta model)

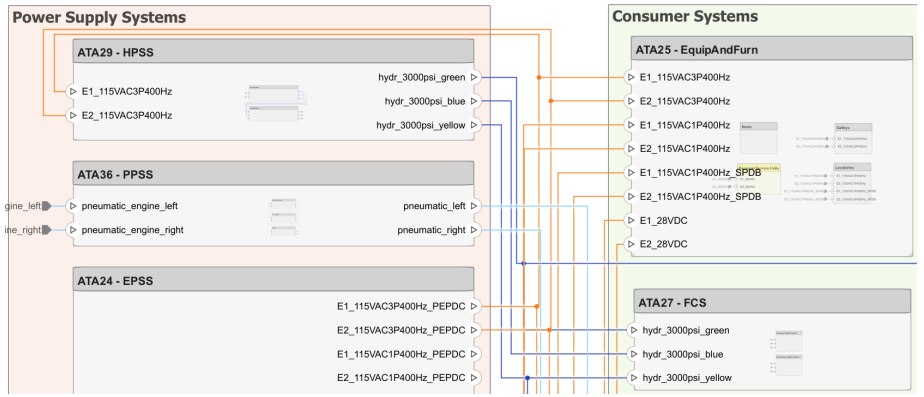
- ○ In-house tool
- – No graphical architecture topology
- – Focus on detailed avionics architectures



System Composer

- ✓Direct interaction with *MATLAB* environment
- ✓*GeneSys* also in *MATLAB*
- ✓Individual architecture views

- – Uni-directional connections
- – No standardized modeling method



Detailed scoring of the four selected MBSE tools

Criteria (weighting factor)	*Capella*	*Cameo System Modeler*	*System Composer*	*Avionics Architect*
• Abstraction capability (3)	5	5	4	4
• Complexity handling (3)	4	4	4	4
• Variants handling (1)	1	2	5	4
• Traceability (3)	5	5	5	3
• Modeling standardization (2)	5	5	1	5
• Graphical interface (2)	3	4	4	2
• Model scalability (1)	4	4	4	5
• Model reusability (2)	4	4	4	5
• Model modifiability (2)	4	4	3	3
• Concurrent modeling (1)	0	0	0	0
• Analysis capability (3)	2	2	4	4
• Constraint checker (1)	2	5	1	0
• Automation capabilities (1)	2	2	5	5
• Future prospect (1)	4	4	4	2
• User guidance (2)	5	1	2	1
• Multidisciplinary modeling (2)	4	4	4	2
• Interaction with *iDAM* (3)	2	4	5	3
• Interaction with *GeneSys* (3)	2	2	5	3
• Total	123	127	136	115