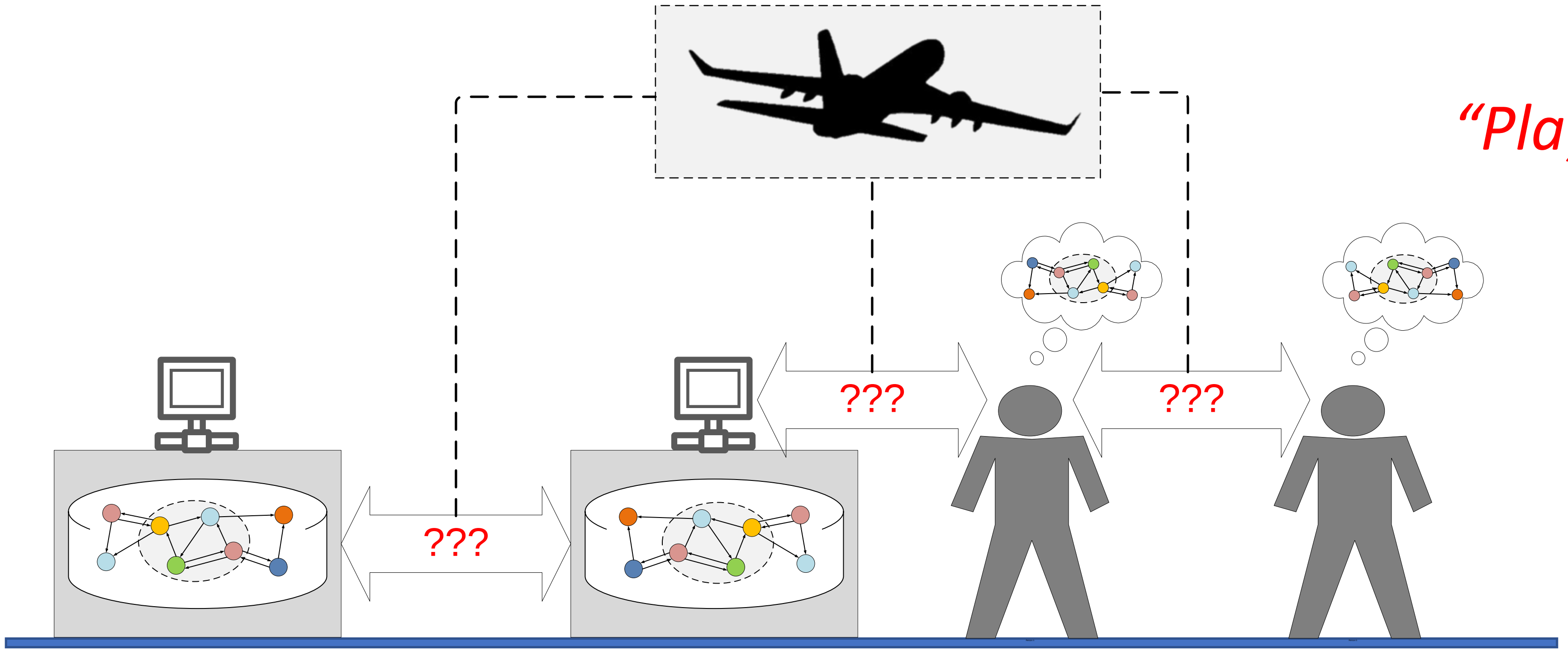
Interoperable Model Semantics: An issue that won't go away



"Play well together"

Bill Schindel
schindel@ictt.com

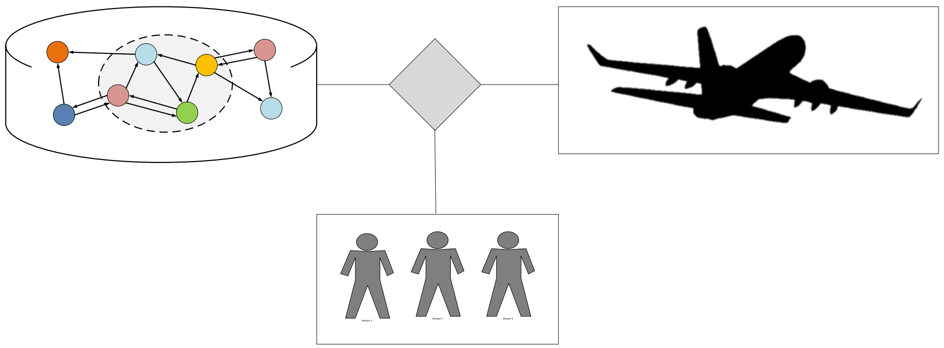
V1.3.1

ICTT System Sciences®
Understand your systems.

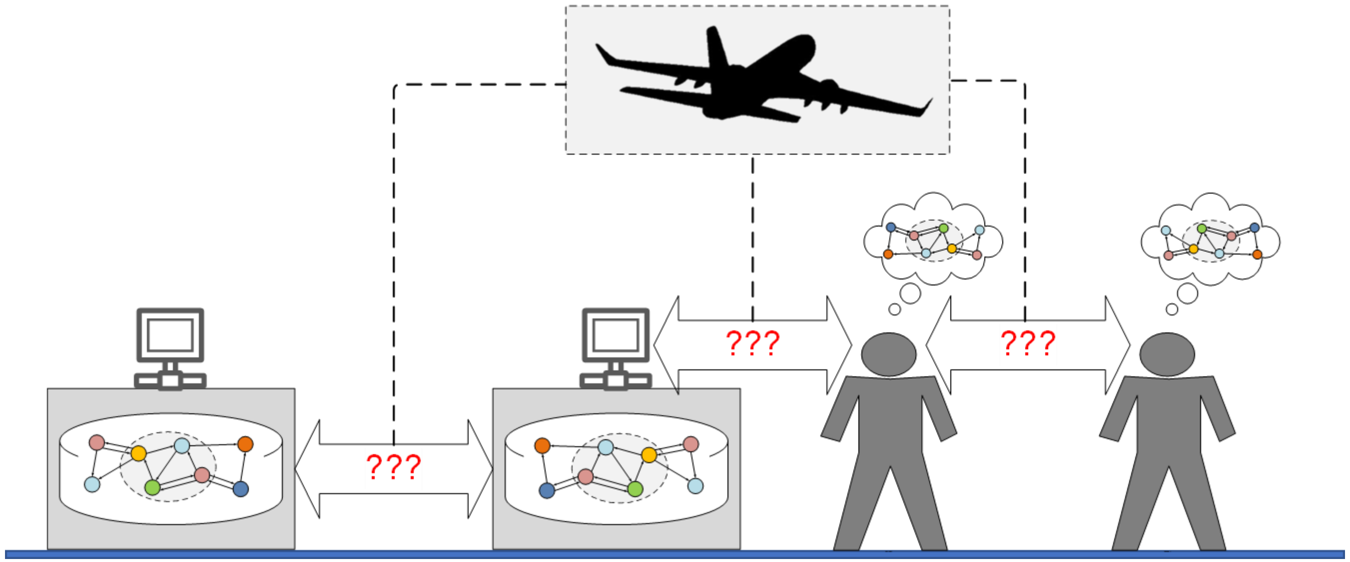
Contents

- Introductions, Background, Setting
- The Need/Problem
- How, When, and Where to Solve It
- Examples
- Questions and Discussion
- References

Introductions, Background, Setting



- Introductions, backgrounds, interests
- The setting for this discussion:
  - Generation and use of computer-based virtual models continues to grow across industry.
  - Likewise, humans also internalize their own mental “models” of systems of interest.
  - Needs for “Interoperability” across all these models can be a challenge.
  - This briefing summarizes the underlying nature of that challenge and its solutions.
  - Some of these aspects are well-known, but certain important aspects sometimes missed.



The Need/Problem (following pages)

- “Playing well together”
- Picking the right problem
- “It is just semantics”
- The web of meaning
- Related efforts and constructs
- Key aspects often overlooked

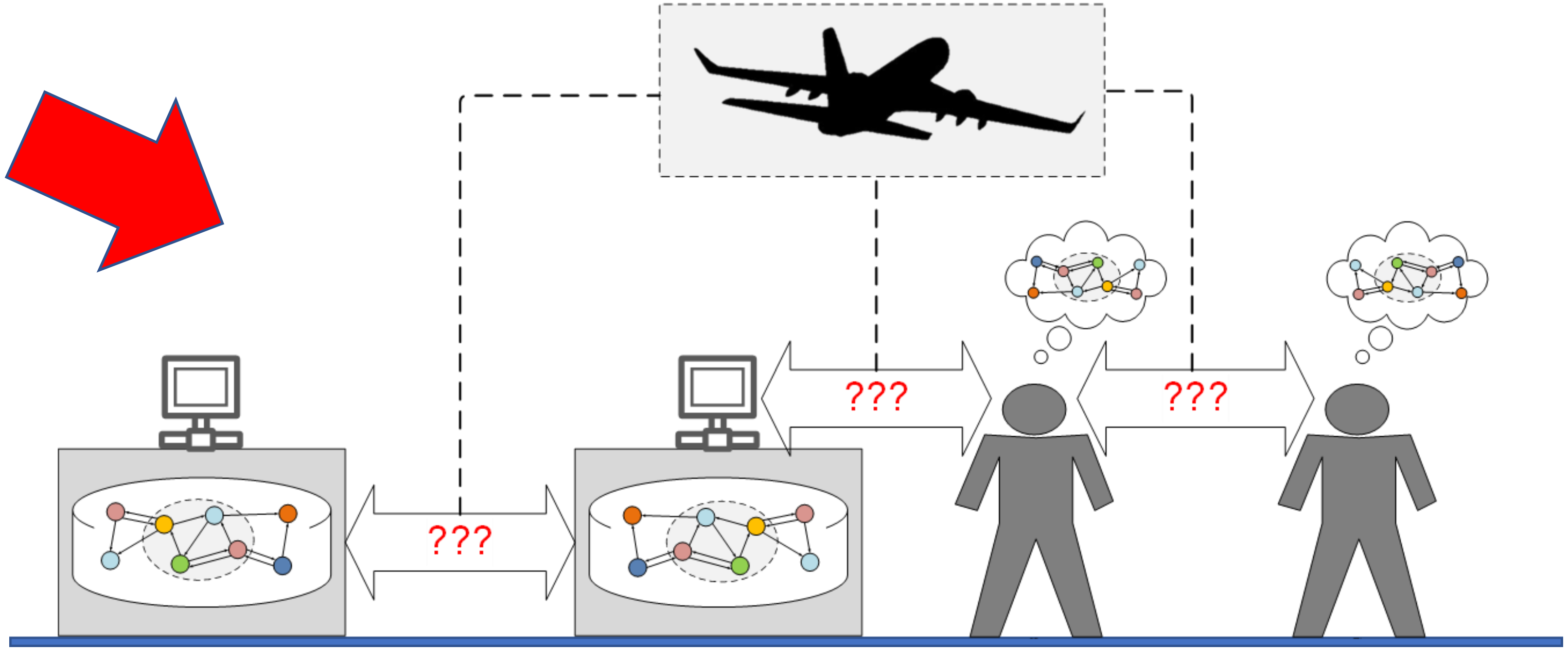
“Playing well together”

- Impact on ability to “work together effectively”, across different individuals, teams, organizations, specialties, capabilities, businesses, etc.
- Includes crossing boundaries across not just between automated tools, but also between different humans individuals/teams, and between tools and people.

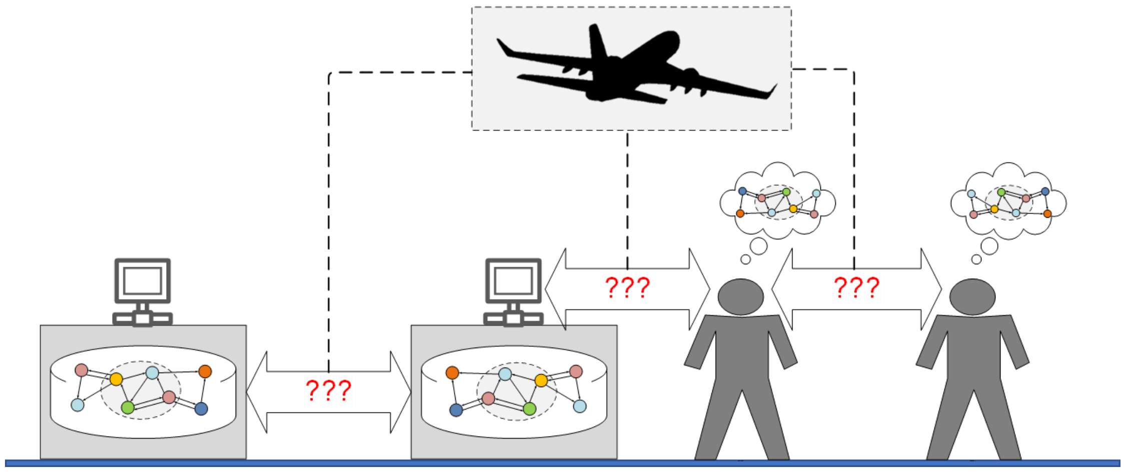
We suggest that the real implications of this simplistic diagram are not as well understood as needed.

What does it really mean?

Why does it matter?



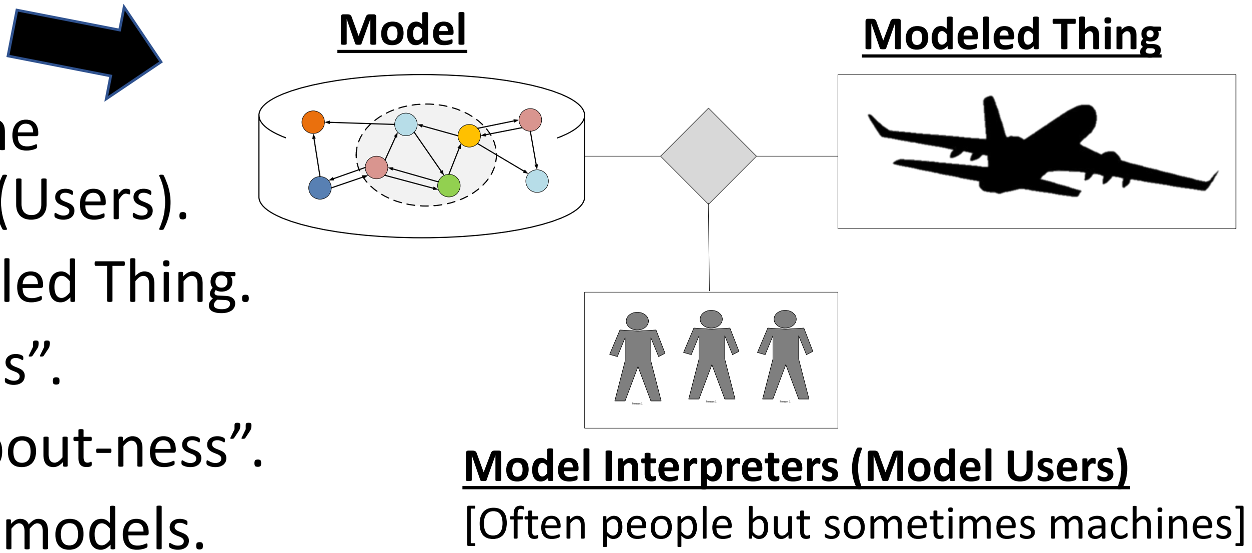
Picking the right problem



- We often hear complaints that various model-based tools and database platforms are “incompatible” with each other, in the sense that they cannot do some kind of “data exchange”.
- This is not a very helpful way to understand the challenge, as it appears to suggest some kind of IT problem that should be solved by automation vendors.
- Simple example: I have a question that requires use of information from several tool chain components. How hard is it for me to query for an answer? Do I have to do the integration in my head, or do the tools help me?
- Bigger picture: What is the overall need? What is the problem of interoperability?

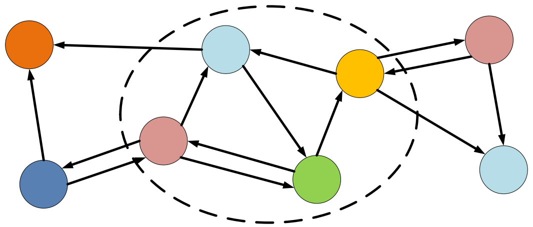
“It is just semantics”

- The modeling community refers to “semantics” of models:
  - In the context of interoperability, what does “semantics” mean?
  - In the common discourse of other contexts, we often hear: “Oh, that is *just semantics*.”
  - This seems to imply some kind of “hair splitting” and unimportance.
  - Also not helpful: Experts will also explain that “semantics” means *meaning*. Whew!
  - What is going on here? What is meant by “semantics” of models?
- This diagram is the right place to start:
  - The Model is said to describe aspects of the Modeled Thing to the Model Interpreters (Users).
  - The Model is said to be “about” the Modeled Thing.
  - We need to understand model “about-ness”.
  - Model semantics is what encodes that “about-ness”.
  - This idea is universal to all types of virtual models.
  - What does model semantics look like? . . . .



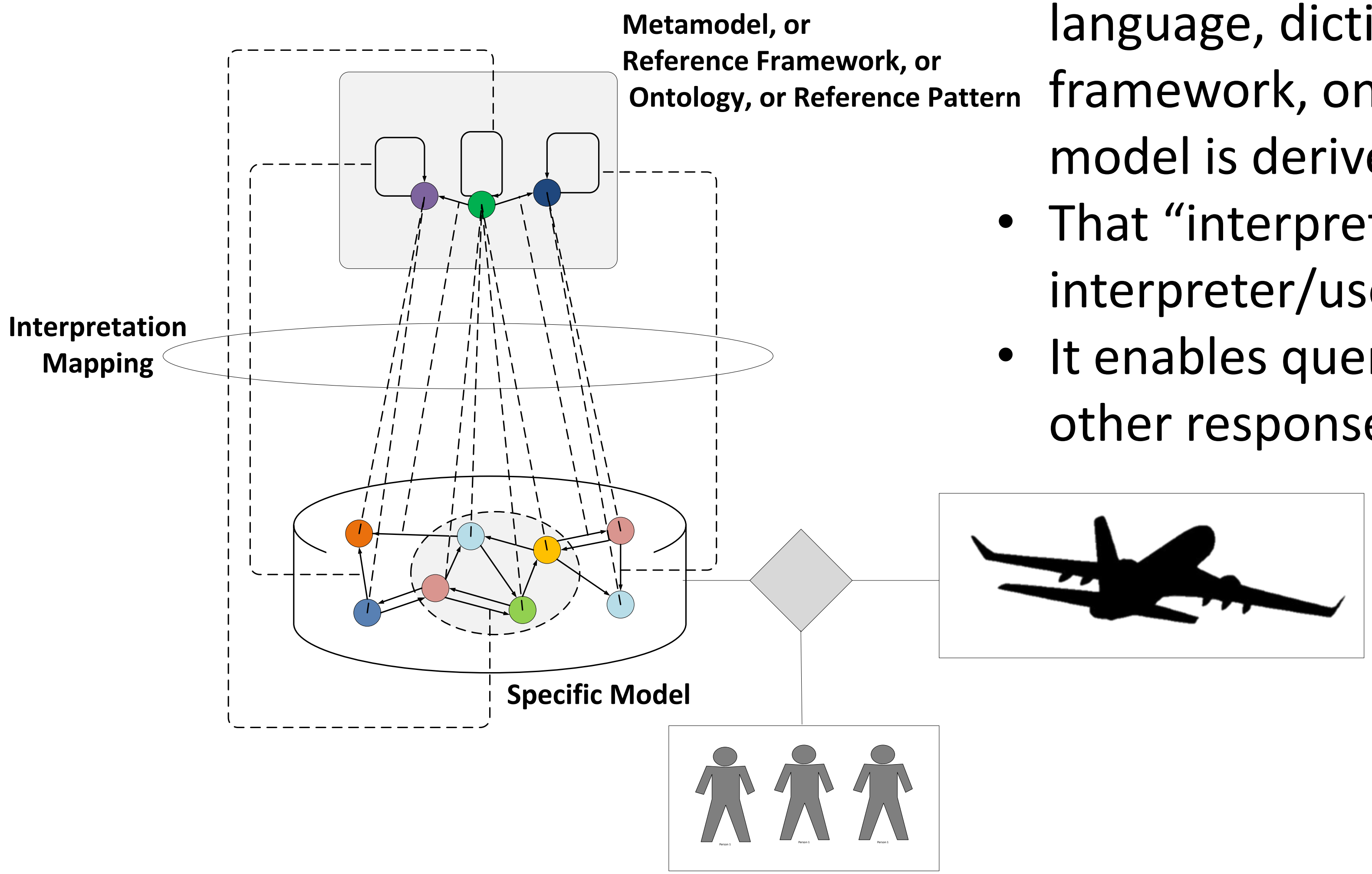
The web of meaning

- Different model types, languages, and tools differ in detail, but . . .
- The core "semantics" of all those models are effectively encoded as information:
  - A web of relationship links (connections)
  - The nodes that are connected
  - The nodes and relationships can have names (including values in some cases).
- The names seem to attract more attention, but are not really the main challenge of model semantics.
- The essence of the semantics are expressed by:
  - The shape of the web of links in the model, and . . .
  - The trace of those nodes and links of a model to a more general "reference" web that expresses the language, dictionary, template, architectural framework, ontology, or pattern from which the model is derived or expressed.



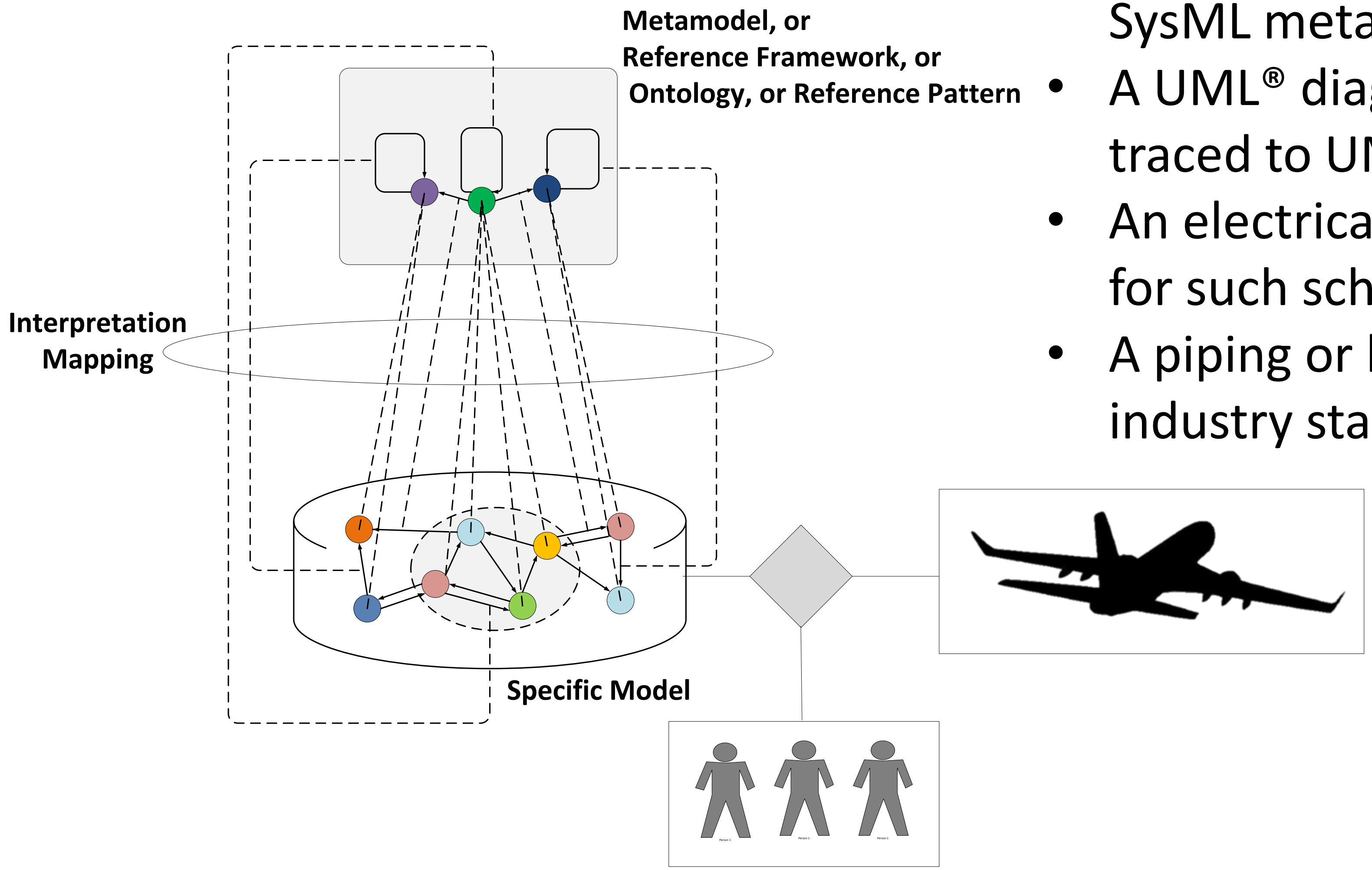
The essence of the model's semantics are expressed by:

- The web of named nodes & links in the model.
- The trace of those nodes and links of a model to a more general "reference" web that expresses the language, dictionary, template, architectural framework, ontology, or pattern from which the model is derived or expressed.
- That "interpretation" trace tells model interpreter/user how to read/interpret the model.
- It enables queries, traces, views, reasoning, logic, other responses.



Examples:

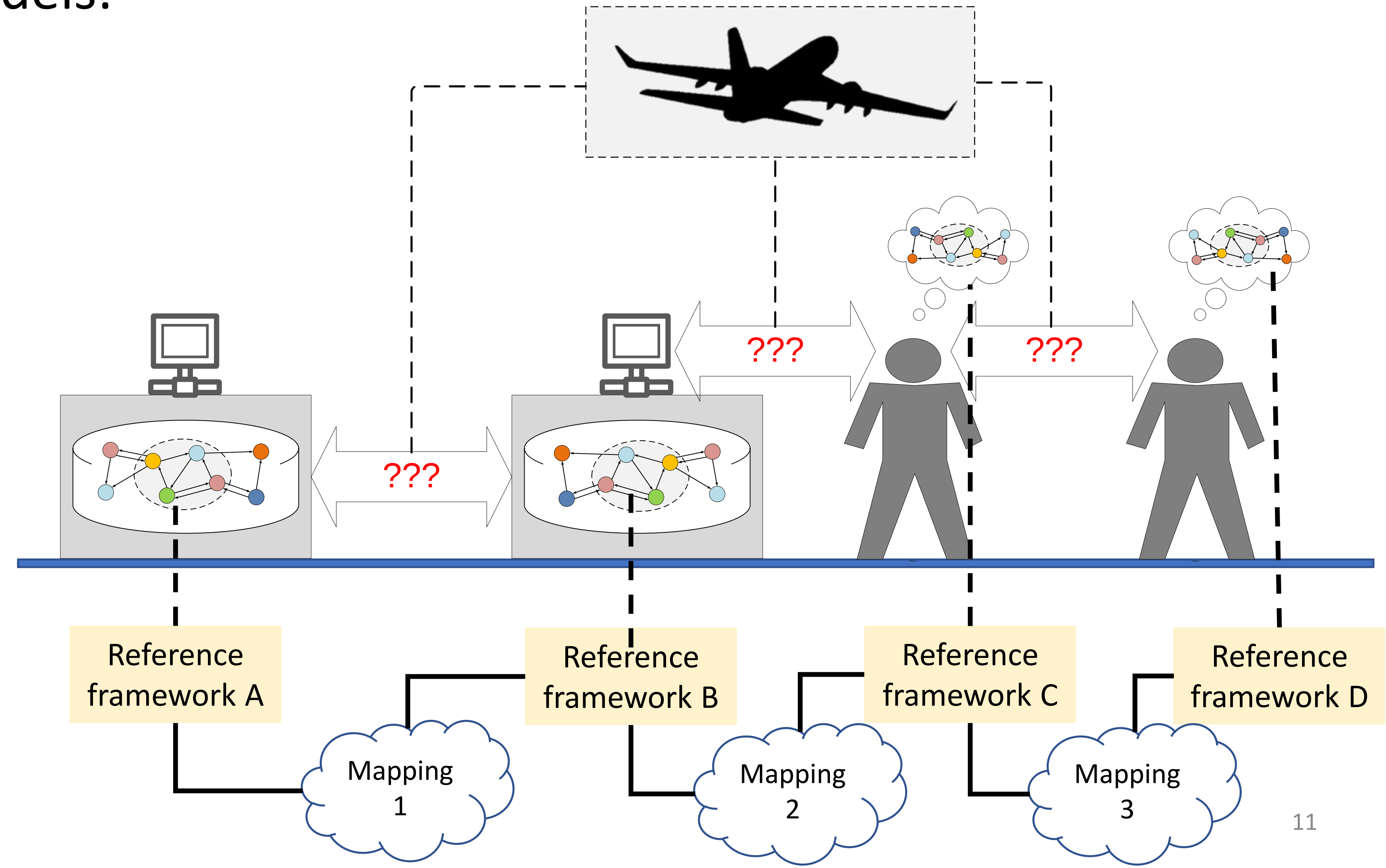
- A Simulink™ simulation model diagram, traced to library of Simulink™ computational blocks;
- A SysML™ model diagram, traced to OMG SysML metamodel or OMG UAF schema;
- A UML® diagram about a computer program, traced to UML metamodel library;
- An electrical schematic, traced to IEC standards for such schematics;
- A piping or hydraulic diagram, traced to industry standards for such diagrams.

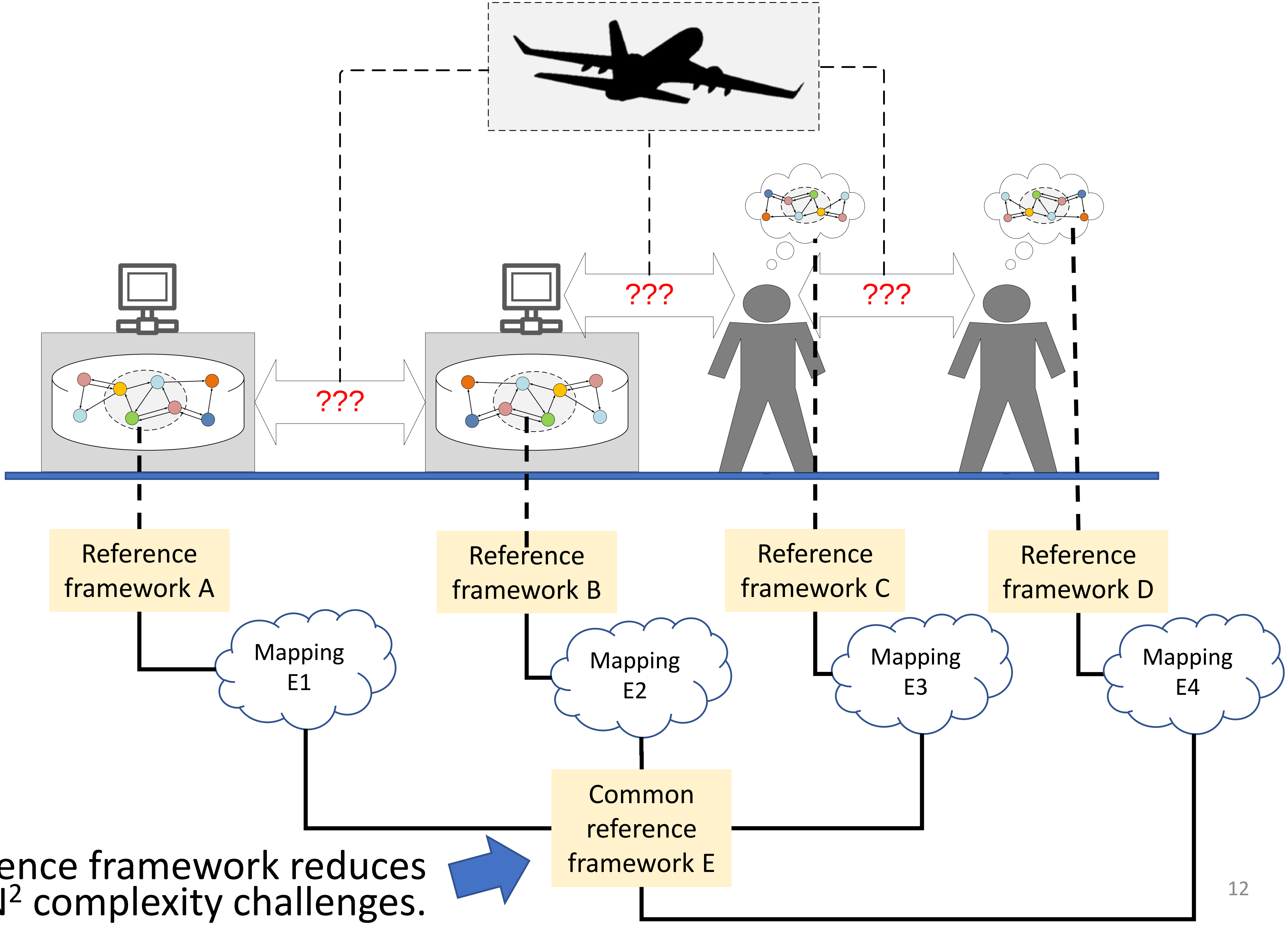


Semantic interoperability

- *Mappings between the reference frameworks*, if they are feasible, are key to interoperability of models:

This diagram understates the typical problem by showing only some "pairwise" mappings. The worst case for that approach is quadratic (N^2) in the number of platforms and people, which is probably overstated.





A “common” reference framework reduces potential N^2 complexity challenges.

Related efforts, constructs, resources: Sampling examples

- Modeling tool language metamodels:
  - OMG SysML Metamodel
  - OPM Metamodel
  - Capella Metamodel
  - Team Center PLM Data Model
- Standards based exchange:
  - ISO 10303 AP233
  - OMG XMI
  - NAFEMS FMI
- Reference metamodels:
  - S*Metamodel
- Reference ontologies
  - Basic Formal Ontology (BFO)
  - Gene Ontology
  - NASA IMCE Ontology
- Hub interoperation & transport tooling:
  - Phoenix Integration ModelCenter
  - Intercax Syndeia
  - HPC Science Gateways (I.U., Apache Airavata)
- Architectural frameworks and related standards:
  - OMG Unified Application Framework (UAF)
  - ISO 42010
- Reference Patterns, Domain Specific Languages (DSLs):
  - ASELCM Pattern
  - Model Characterization Pattern (Wrapper)
  - Manufacturing Pattern
  - Embedded Intelligence (EI) Pattern
  - Product Line Engineering (PLE) Models
- Schemas:
  - Navy Federated Schema

Model incompatibility vs. Model insufficiency

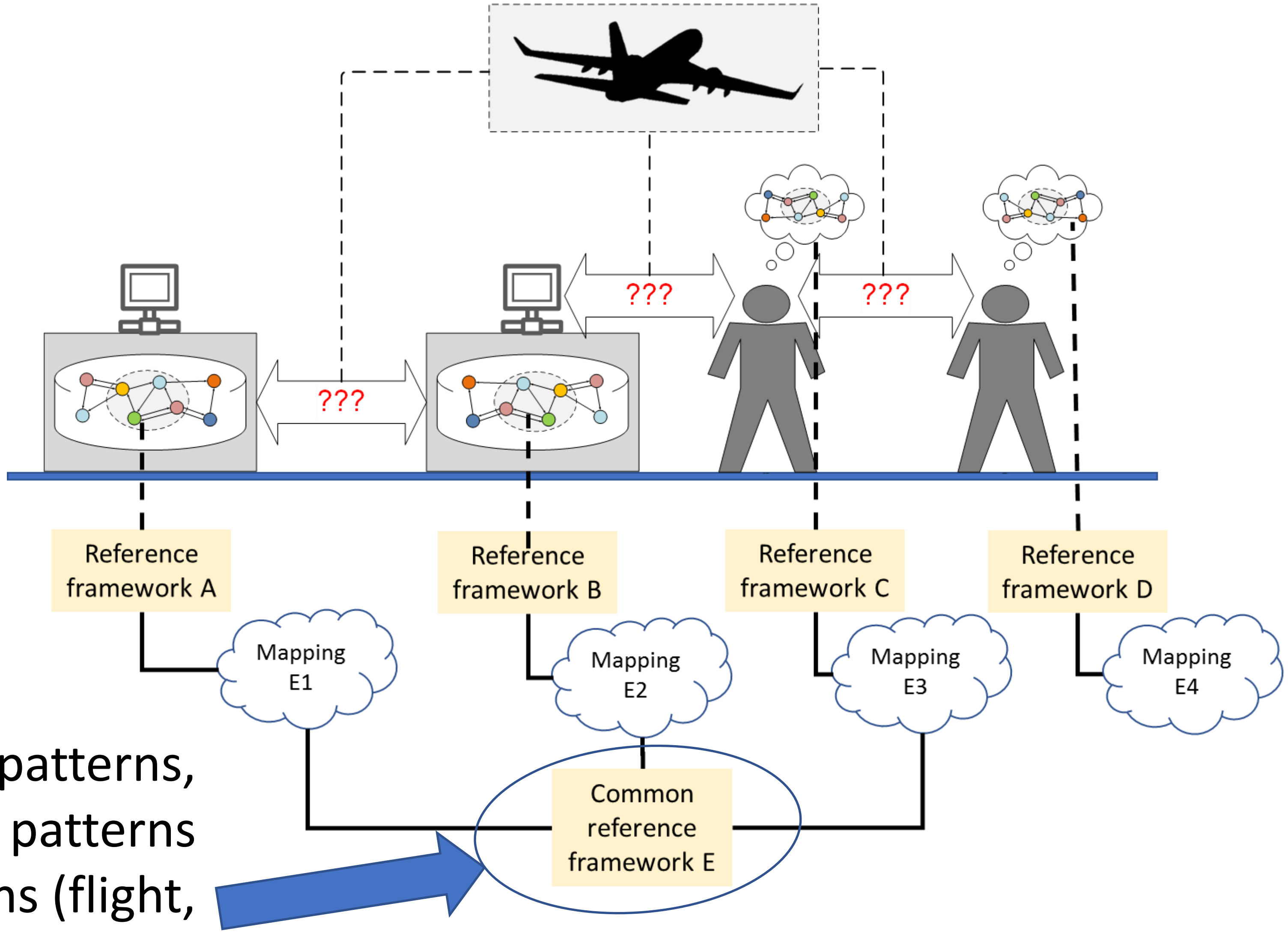
1. **Reality Laboratories**: Physical science (and engineering) for ~300 years has dramatically lifted our lives by pressing physical science models to agree with Nature.
2. **Consensus 'Committees'**: Commerce and information technologists for ~50 years progressed by pressing its information models to agree with each other.

- The above (both needed) are not necessarily in conflict, but sometimes do conflict:
  - See Galileo versus The Inquisition (models of the earth-sun-planetary system).
  - "Conventional Wisdom" is sometimes wrong, even if agreeable.
- More recent emphasis on formalized consensus standards for systems modeling languages has in some cases disconnected systems engineering models from (1):
  - It is very possible (and currently the case) for standards-based tools to be consistent with each other (interoperable) but inconsistent with the Nature (not so useful) they are expected to represent.
  - Example: Interactions (contrast to computational model emphasis on fidelity in representing phenomena of Nature).
  - These are currently resolved by mappings that formalize consistent representation of missing elements in the same existing third party COTS tools.

New levels bring new recurrences

- As higher level system domains (e.g., aircraft engines, whole aircraft, multiple aircraft-satellites-ground stations) are modeled, larger scale, higher-level interactions appear, and from these emerge new semantic patterns.
- This is exactly identical to the emergence of chemistry from atomic physics, and the emergence of biology from chemistry.
- It is not true that establishing lower-level semantic interoperability solves the interoperability of the higher-level domains.
- A similar misunderstanding in physics led to a famous paper (“More Is Different”) by Nobel Laureate P. W. Anderson.
- This means we are never “done” creating semantic interoperability! The ability to do so must be practiced as an ongoing part of innovation.

New levels bring new recurrences



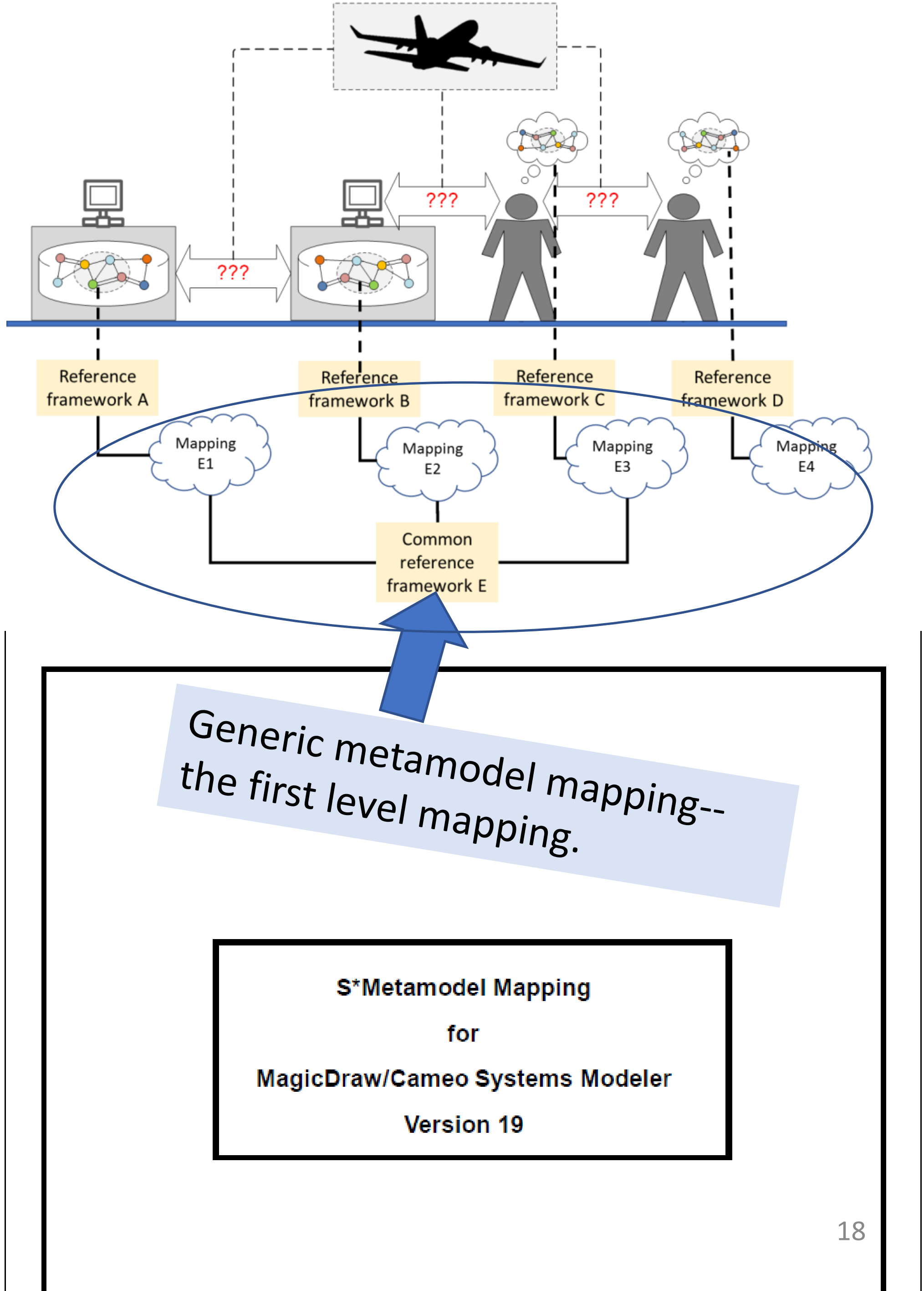
Becomes a hierarchy of reference patterns, from general systems to intermediate patterns (e.g., cybersecurity) to specific domains (flight, medical, automotive, manufacturing, etc.)

How, When, and Where to Solve It (following pages)

- What you can do right now
- Reducing proliferation
- Group learning—essential for playing well together

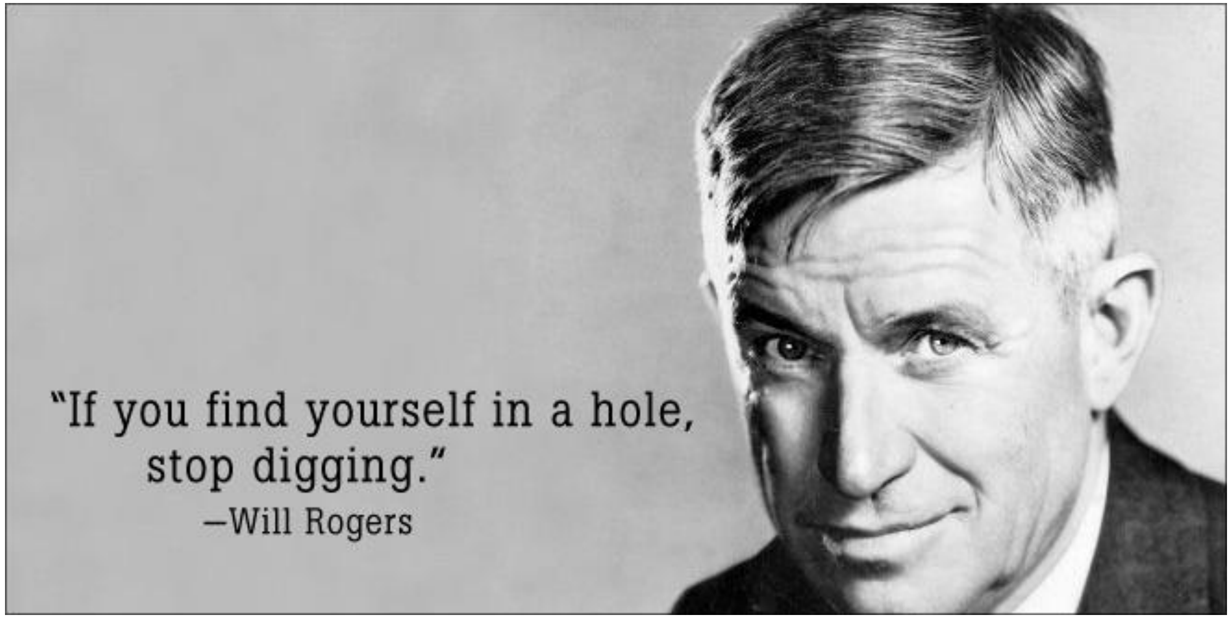
What you can do right now (and should do first anyway)

- Use mappings of your tools and information systems to a common reference framework
- This does not require buying new tools, writing new programs, etc.
- It does require humans to work together—after all, “playing well together” was what we set out to study in this briefing!
- This establishes the basis of semantic interoperability, without getting into other automation issues until you are ready to do so later.
- Example: Computational Model / Context Model
- You can do this now, if you have the will; if you don’t, programmers and software packages won’t help much later.
- Map to S*Metamodel, domain specific S*Patterns, etc.



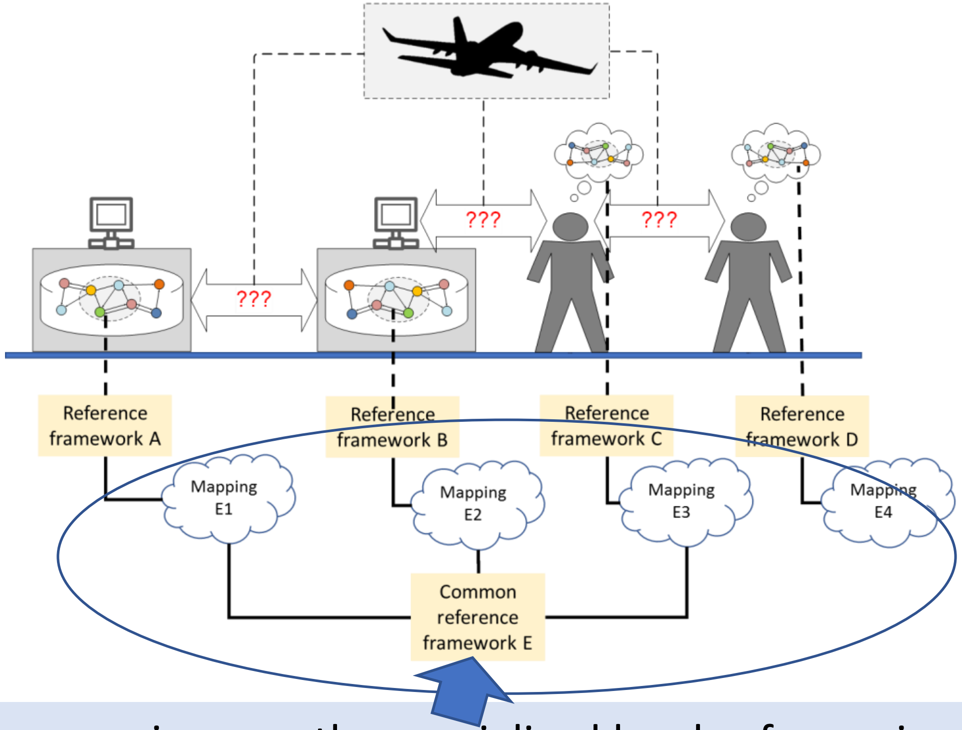
Reducing proliferation

- Begin applying governance to generation of new models, by mapping them to common references.
- Even when some exceptions to this are tolerated, this begins reducing proliferation of additional inconsistency complexity at the N^2 level.
- Every variant does not require a separately originated model.



Group learning: Essential for “playing well together”

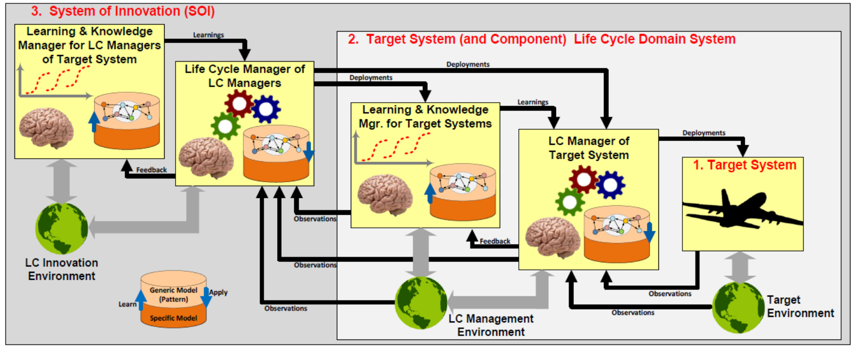
- The higher level domain mappings are in effect discovery of recurring patterns that can be “configured” for individual cases.
- This is exactly the “Group Learning” discussed in the related reference session shown below.
- In an innovation organization, Group Learning is the essential core of “Playing Well Together”.
- Uncover the Pattern™ (UTP) is a fast way to make this happen.



Domain pattern mappings are the specialized levels of mapping.

Uncover the Pattern: Harnessing Group Learning for:

- Integrating Improvements to Engineering & Life Cycle Management Capabilities
- Integrating Improvements to Product Capabilities



ICTT System Sciences®
Understand your systems.

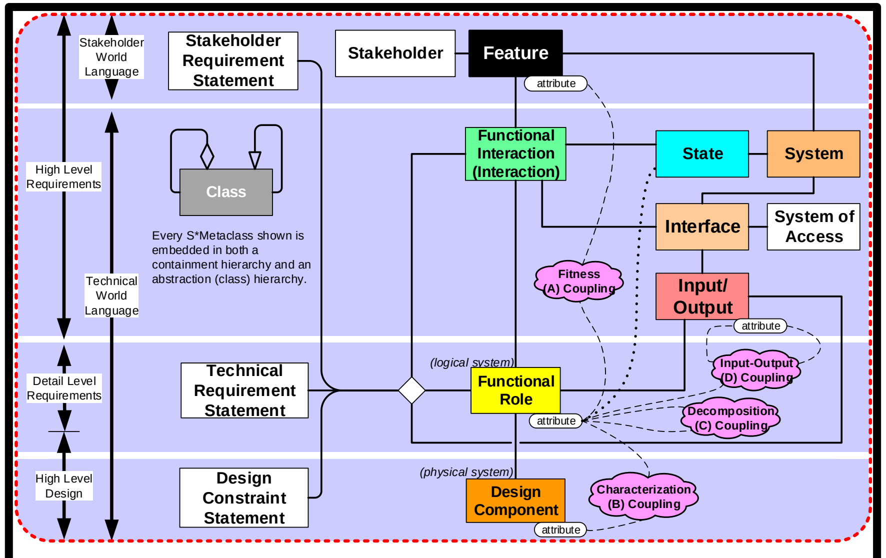
Bill Schindel
schindel@ictt.com
04.06.2020 V1.3.1

Examples (following pages)

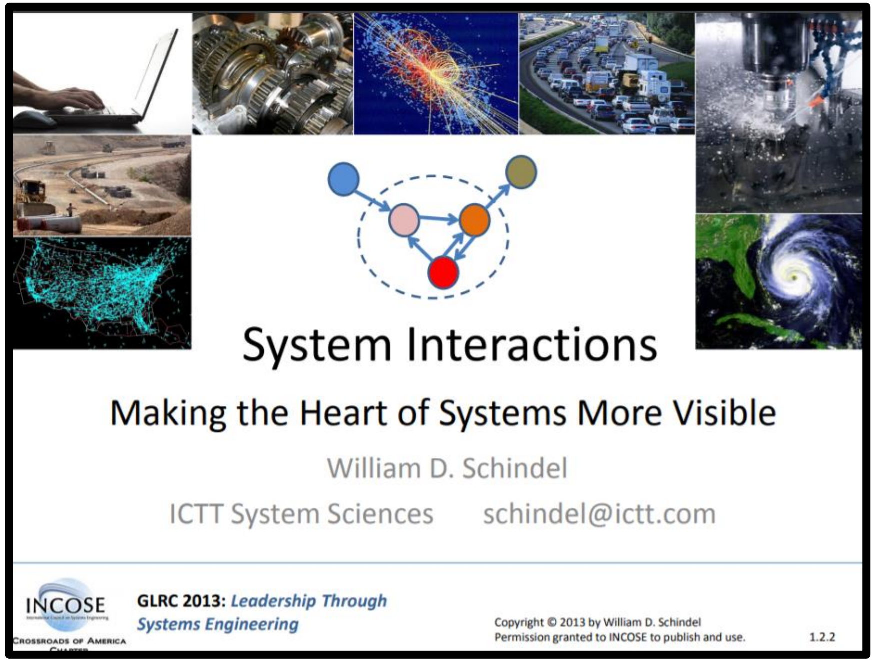
- S*Metamodel and System Interactions
- Model Characterization Pattern (MCP, Model Wrapper)
- Computational Modelers vs. Systems Modelers: Context Models
- Embedded Intelligence Pattern (Parent of Cybsersecurity Pattern)
- Mapping of S*Metamodel to Magic Draw / Cameo Systems Modeler SysML
- Foundations of Systems Engineering

S*Metamodel and System Interactions

- Mappings to the S*Metamodel establish basic semantic interoperability at the level of generic systems.
- Part of the S*Metamodel is about Interactions.
- For example, such mappings enable any third party COTS tool or information system to be S*Model semantics capable.
- See the References both these related documents.

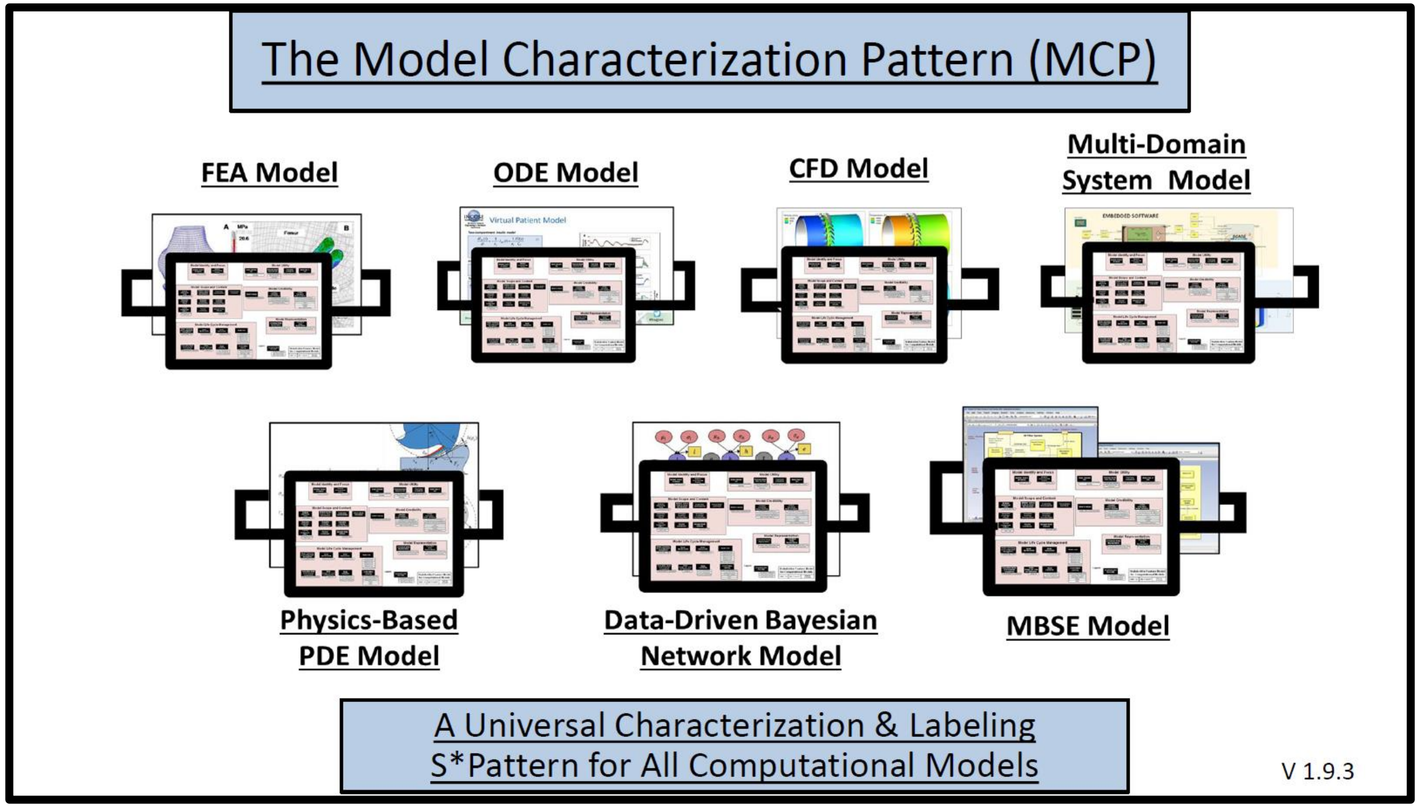


S*Metamodel informal summary pedagogical diagram
(formal S*Metamodel includes additional details.)



Model Characterization Pattern (MCP) (AKA Model Wrapper)

- Even for models that are semantically incompatible across groups or tools, the "model wrapper" pattern provides universal metadata which describes any model.
- So, it can be applied immediately to existing or new models, *even without taking the initial steps to make them semantically compatible.*

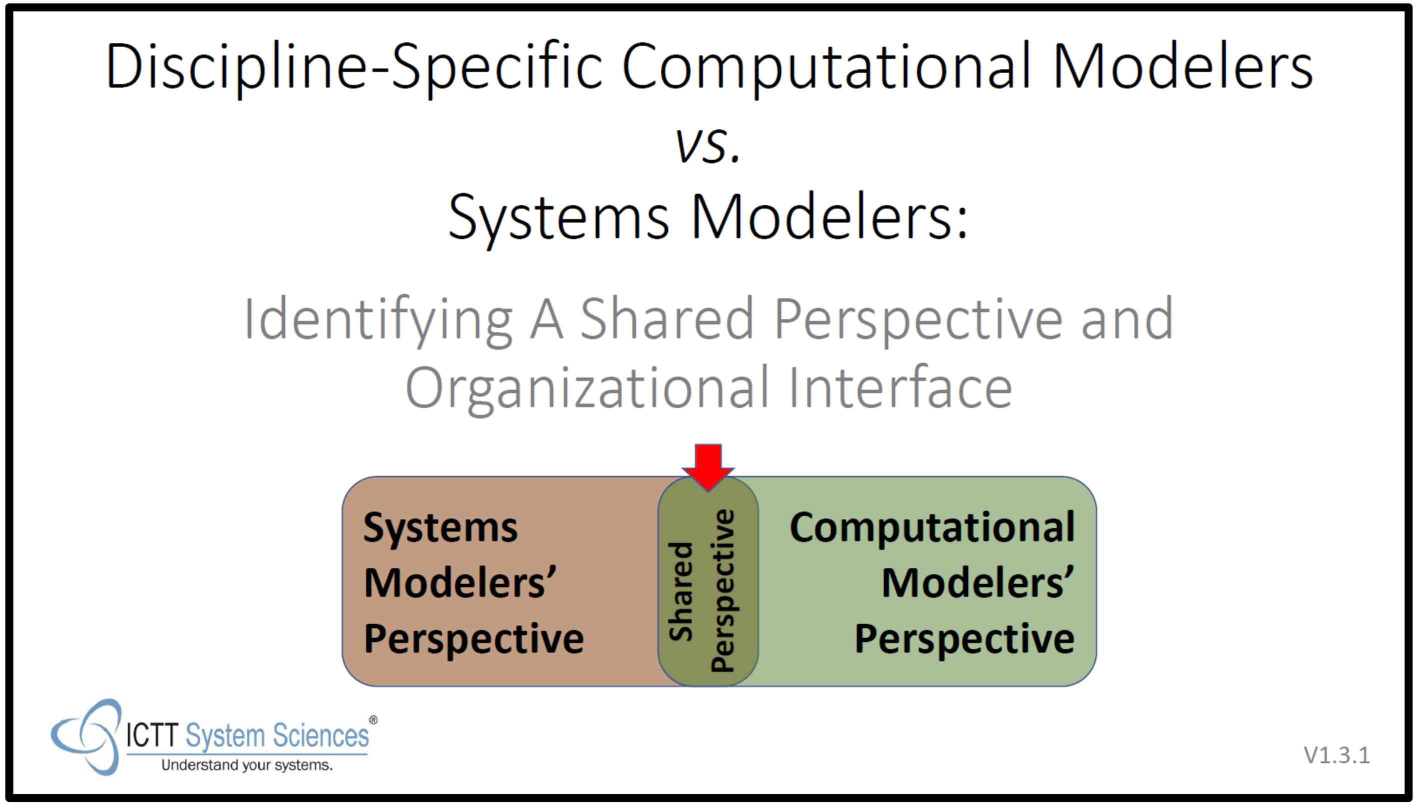


https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:model_characterization_pattern_--_summary_guide_v1.2.1.pdf

https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:model_characterization_pattern_mcp_v1.9.3.pdf

Computational Modelers vs. Systems Modelers

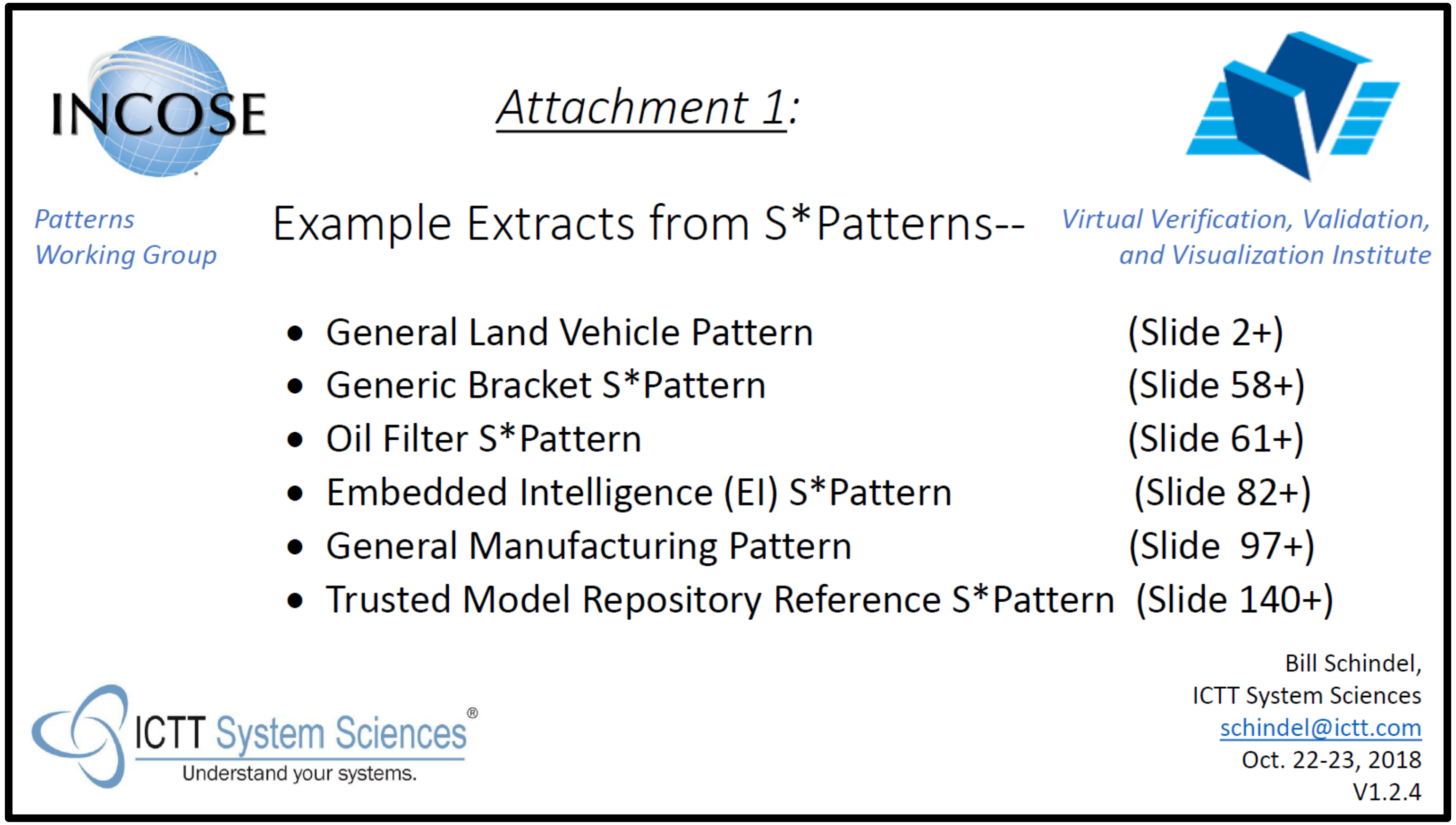
- Often the mappings between different groups or tools needs to address only a subset of information types that is relevant to each group.
- This is well-illustrated by the mapping between computational models in general (simulations) and systems (MBSE) models in general.
- This resource summarizes the two perspectives and the subset of the S*Metamodel that applies in such a mapping.
- It illustrates that it is not necessary to force the people involved to see any more than they need.
- It introduces the System Context Model for a Computational Model, as an aid for both, providing a mapping in the form of a model.



https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:common_ground_seam_v1.3.1.pdf

EI Pattern (Parent of Cybersecurity Pattern)

- The Embedded Intelligence (EI) Pattern is the parent from which the Cybersecurity Pattern can be derived.
- As an S*Pattern, the EI Pattern can be inherited into other S*mapped models, creating a common configurable framework for embedded intelligence in general and cybersecurity in particular.



INCOSE

Patterns Working Group

Attachment 1:

Example Extracts from S*Patterns--

Virtual Verification, Validation, and Visualization Institute

- General Land Vehicle Pattern (Slide 2+)
- Generic Bracket S*Pattern (Slide 58+)
- Oil Filter S*Pattern (Slide 61+)
- Embedded Intelligence (EI) S*Pattern (Slide 82+)
- General Manufacturing Pattern (Slide 97+)
- Trusted Model Repository Reference S*Pattern (Slide 140+)

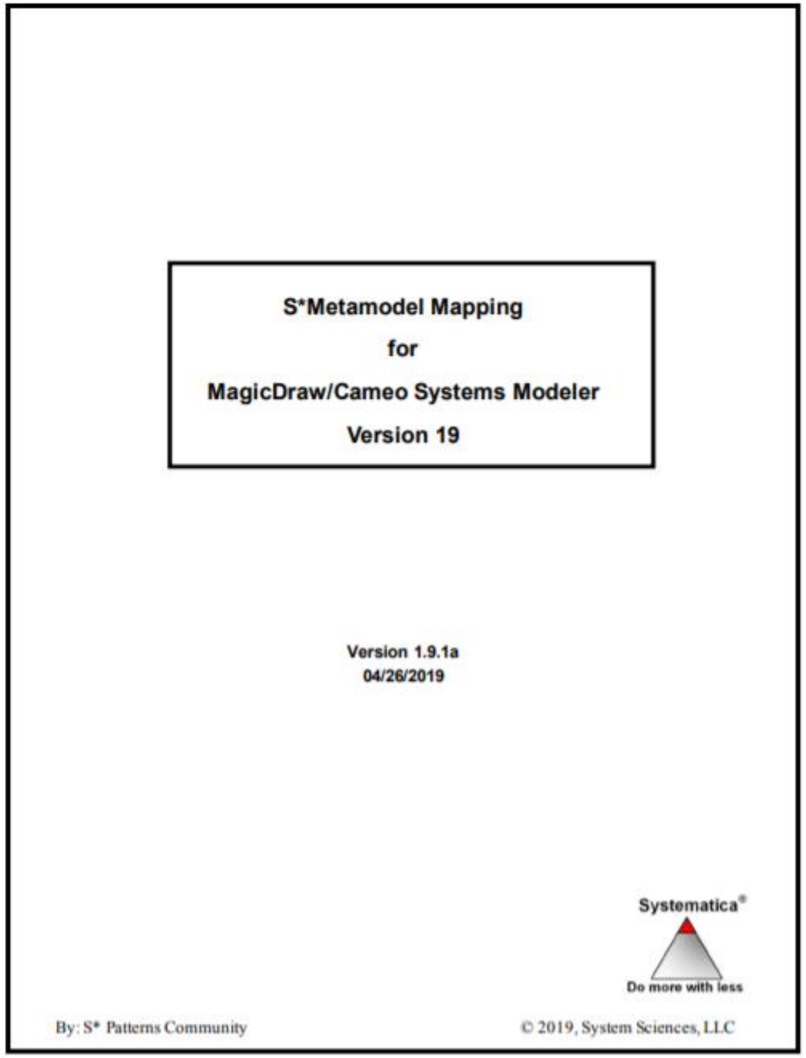
ICTT System Sciences®
Understand your systems.

Bill Schindel,
ICTT System Sciences
schindel@ictt.com
Oct. 22-23, 2018
V1.2.4

https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:attachment_1--example_extracts_from_s-patterns_v1.2.4.pdf

Mapping of S*Metamodel to Magic Draw / Cameo Systems Modeler SysML

- Illustrates a typical mapping to a third party COTS modeling tool schema—in this case, a SysML modeling tool.
- Establishes basic semantic interoperability for the tool with other S*Metamodel mapped tools, information systems, or humans.


S*Metamodel Mapping

for

MagicDraw/Cameo Systems Modeler

Version 19

Version 1.9.1a

04/26/2019

Systematica®

Do more with less

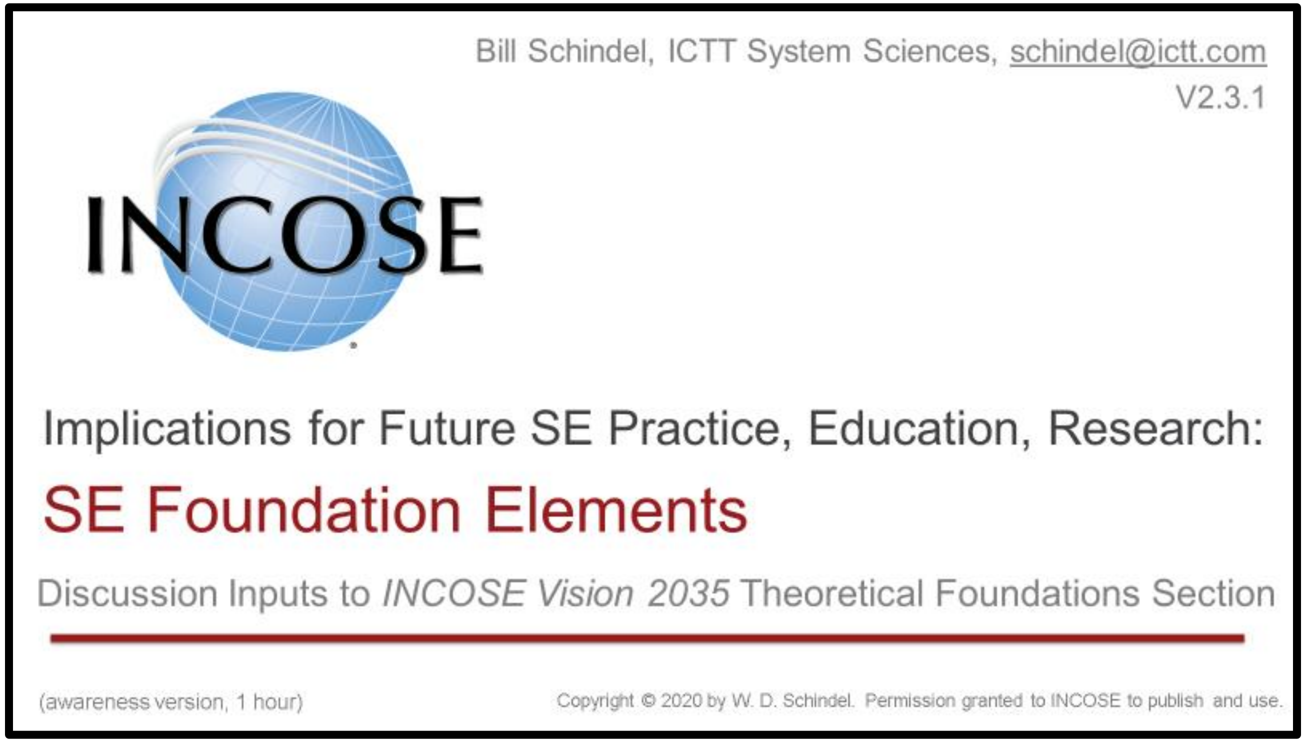
By: S* Patterns Community

© 2019, System Sciences, LLC

https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:systematica_mapping_for_magicdraw_csm_v1.9.1a.pdf

Foundations of Systems Engineering

- It is the underlying nature of interacting higher level systems that generates new ontologies and requires additional mapping discipline to avoid interoperation problems.
- See Slides 20-22, 69.



https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:science_math_foundations_for_systems_and_systems_engineering--1_hr_awareness_v2.3.1a.pdf

Questions, Discussion

-
-
-
-
-
-
-

References

1. "What Is the Smallest Model of a System?", in Proc. of INCOSE 2011 International Symposium, 2011. https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:what_is_the_smallest_model_of_a_system_v1.4.4.pdf
2. "Interactions: Making the Heart of Systems Visible", in Proc. of the INCOSE Great Lakes 2013 Regional Conference", 2013. https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:system_interactions--making_the_heart_of_systems_more_visible_v1.2.2.pdf
3. Formal S*Metamodel https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:systematica_5_metamodel_v7.1.6a.pdf
4. "Patterns Across the Enterprise", in Proc. of the INCOSE 2015 International Symposium", 2015. https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:patterns_across_enterprise--life_cycle_mgmt_is2015_v1.4.2.pdf
5. INCOSE MBSE Patterns Working Group Web Site: https://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns
6. Schindel, W., and Dove, R., "Introduction to the ASELCM Pattern", Proc. of INCOSE IS2016, Edinburg, UK, 2016 https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:is2016_intro_to_the_aselcm_pattern_v1.4.8.pdf
7. http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:pbse_extension_of_mbse--methodology_summary_v1.5.5a.pdf
8. http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:pbse_tutorial_glrc_2016_v1.7.4.pdf
9. http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:mbse_patterns--public_private_and_hybrid_schindel_v1.2.3.pdf

For readers with interest in more information, see the above INCOSE/OMG MBSE Patterns WG web site, and also the slide "Related efforts, constructs, resources: Sampling examples" for many other references, not listed immediately above.