Identification and Mitigation of Security Classification Marking Challenges and Risks for Descriptive Models

Ryan Noguchi
The Aerospace Corporation

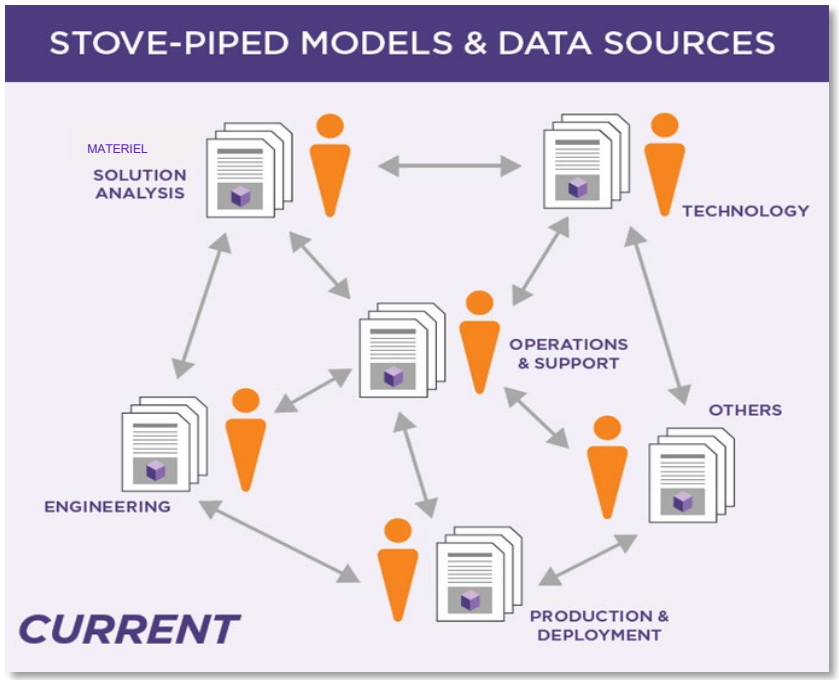
NDIA Systems and Mission Engineering Conference
October 2023

Approved for public release. OTR-2023-01060.

Transitioning to Digital Engineering

Document-driven engineering

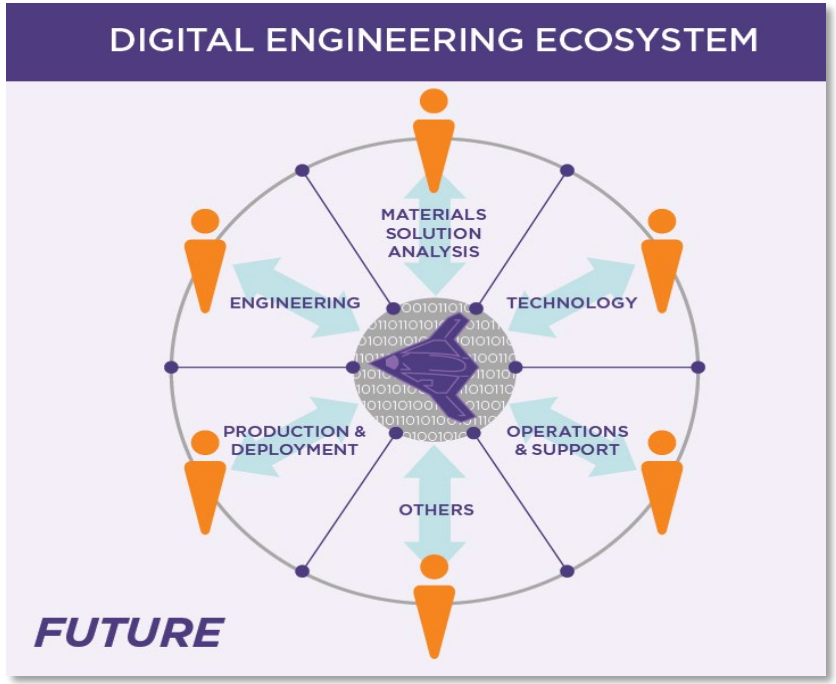
Document-Centric Culture



- Knowledge is embodied in static, disconnected artifacts

Digital engineering

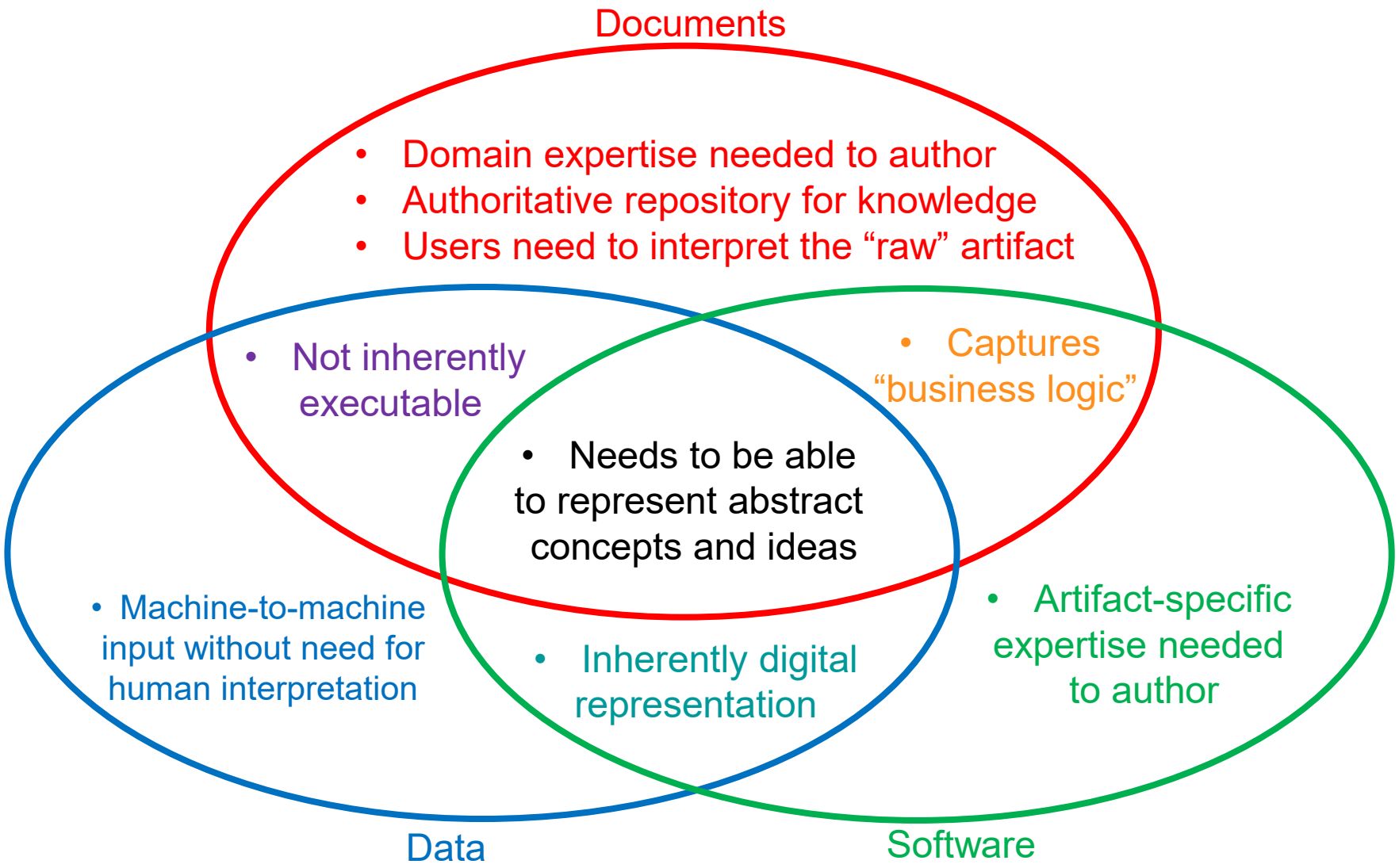
Data/Model-Centric Culture



- Knowledge is embodied in digitally connected models

SE must transition to MBSE to enable this DE transformation

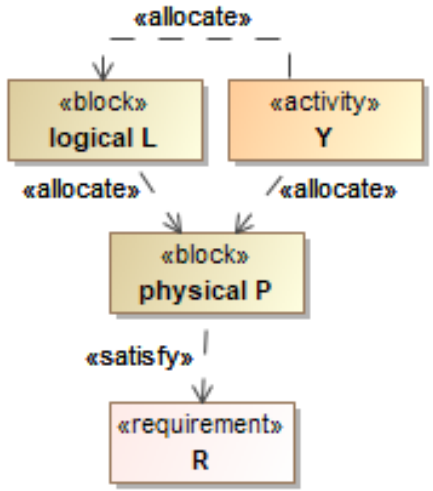
Introduction – MBSE and the characteristics of descriptive models



Descriptive models have similar characteristics to documents, data, and software

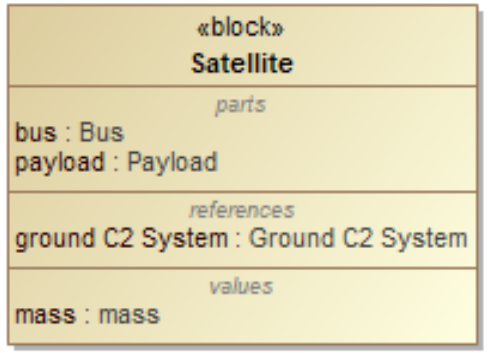
What is a Model, Really?

- MBSE is the practice of SE in which models replace documents as the embodiment of SE knowledge
  - *These models capture the information that was previously captured in documents*
  - *These models have more rigorous rules for their implementation and interpretation than natural language*
    - Less ambiguous than natural language documents
    - More information-dense than documents



- A descriptive model is essentially a set of assertions, captured in a compact notation
  - *A single model element can contain a lot of information, perhaps many paragraphs of prose*

- The model can include many different classes of assertions
  - *What is true*
  - *What has been determined to be true*
  - *What shall be true—a specification of the requirements*
  - *What should be true—a description of the expectation*
  - *What could be true—what is allowed, or what possibly may be true, some of the time*



To which of these does the term "fact of…" apply?

What is Security Classification?

- Security classification rules specify that a quantum of factual information (fact X) in some context Y should be protected at a level Z
  - *The Original Classifier's job is to create and document these rules*
  - *The Derivative Classifier's job is to apply these rules to their document*
  - *The artifact user's job is to understand what these rules mean and act accordingly*

- To understand what a classification rule means, we need to break it down into its three pieces
  - *Of particular importance and danger is Context Y*

- To model a classification rule appropriately, we need to be clear on what that rule means
  - *Are we modeling the correct Fact X?*
  - *Is the classification rule being made in the correct Context Y?*
  - *Are we adequately reflecting the Level Z?*
    - Particularly in a manner that enables us to find subsets or wholes when we need to?
      - *Sometimes I want both "S//NF" and "S" in a query but other times I only want one of these*

Context is Everything

- Model elements are used in many different contexts
  - *Each model view is potentially a new context*
  - *Each connection to a different model element establishes a new context*
  - *Some of these contexts are not very visible to the human model user*
    - E.g., implied and indirect relationships

- A federated set of models could be construed to be a “compilation” of the information they collectively contain
  - *Each subset of that federation establishes a unique context*
  - *Each accessible subset of that federation establishes a unique context*
  - *Some classification guidance rules classify compilations at a level above any of its individual components*

- Security classification guidance is rarely explicit about the context Y within which the “fact” should be classified
  - *As a result, the default position is that the classification is applied to all contexts*
  - *But then how do you check each possible context for applicability of that classification criterion?*
    - Is it even really practical to check all contexts?

The application of existing document-focused guidance to models is not as straightforward as many assume

Classifying a Model Element—What does that really mean?

- What does it mean when one applies a single classification level Z to a model element? What is classified?
  - *The name of the model element?*
    - Or the name of the entity represented by the model element?
    - What if one name of that entity is classified and another is not?
  - *The existence of the model element?*
    - Or the existence of the entity represented by the model element?
  - *The visibility of the model element?*
    - Or the visibility of the entity represented by the model element?
  - *The placement of the model element within the model organization?*
    - E.g., the model element is contained in some specific package, block, or requirement
  - *Any, all, or some base classifier(s) of the model element?*
    - Not the base classifier itself, but the fact that this element is a specialization of that base classifier
    - Or the fact that the element is a specialization of two or more specific base classifiers?
  - *Any, all, or some usages of that model element?*
  - *Any, all, or some instances of that model element?*
  - *Are the contents of the model element classified?*
    - i.e., is the classification statement akin to the banner line of a document, where the model element is considered to be a container for its contents?

These are all distinct assertions; which are applicable when we apply one classification marking to an element?

Classifying a Relationship—What does that really mean?

- What does it mean when one applies a single classification level Z to a relationship? What is classified?
  - *The name of the relationship?*
  - *The existence of the relationship?*
    - Existence from whose perspective? (From one or the other element at the ends, or a third party?)
  - *The role at one end (or both ends) of the relationship?*
  - *Any, all, or some properties of the relationship?*
  - *Any, all, or some attributes of the relationship?*
  - *Any, all, or some base classifier(s) of the relationship?*
    - Not the base classifier itself, but the fact that this relationship is a specialization of that base classifier
    - Or the fact that the relationship is a specialization of two or more specific base classifiers?
  - *The definition of the relationship or its usage?*
- Is the relationship's classification based on its usage in a diagram?
- Is that classification specific to that diagram?

These are all distinct assertions; which are applicable when we apply one classification marking to an element?

Classifying a Value Property—What does that really mean?

- What fact regarding the value property is classified?
  - *The name of the value property?*
  - *The existence of the value property?*
  - *The existence of the value property as an attribute of this specific element definition?*
  - *The existence of the value property as an attribute of this specific usage of this element definition?*
  - *The default value of the value property?*
  - *The initial value of the value property?*
  - *The redefined value of the value property?*
  - *The current value of the value property?*
  - *Any value of the value property?*
  - *The range of possible values of the value property?*
  - *The multiplicity of the value property?*
  - *Only values of the value property that represent the actual operational value?*
- Note that some of these may not be independently markable at all
  - *How do you mark a numerical value or a multiplicity??*
- If the value is what's classified, generally that's what should be marked, not the value property itself
  - *But how do you mark a numerical value?*

These are all distinct assertions; which are applicable when we apply one classification marking to an element?

Classification Levels—and how they relate

- Classification levels can be decomposed into three parts, some of which are optional:
  - *1) Classification level A*
  - *2) Control marking(s) B1, B2, …, Bn*
    - Order is SCI, then SAP, then AEA, then FGI, separated by double-slashes
    - There may be sub-control or compartment markings containing dashes
  - *3) Dissemination marking(s) C1, C2, …, Cn, separated by double-slashes*
    - Order is DISSEM, then OTHER DISSEM
- Syntax is therefore:
  - *A//B1/B2/…/Bn//C1//C2//…//Cn*
    - Make sure you don't type it out of order
    - Make sure you don't enter Bx or Cx in a different order than the "official" one
    - Make sure you don't have a typo anywhere
    - Make sure you don't use the wrong delimiting characters—can we even use forward slash characters?
- Also, note that banner markings and portion markings are often different
  - *Some abbreviations used in portion markings are technically not allowed in banner markings*

Using text strings to distinguish classification levels is problematic

Implementing Security Markings—Types

- Raw string values are problematic, for the reasons described earlier
  - *How do you query on these strings? Hope you love regular expressions!*
- Enumerations at least eliminate the typo risk, but introduce other challenges
  - *You need a separate enumeration literal for each combination you need to use*
  - *How do you manage enumeration definitions that need to grow as models move up in classification?*
    - Some of those literals may not be allowed at lower classification enclaves
- Reification (into first-class objects) of the security marking concept provides most flexibility but adds complexity
  - *How would we combine the components of the classification marking?*
    - Multiple generalization may be closer to the right semantic than composition
      - *Is "S//NF" a specialization of "S" or vice versa?*
      - *Singleton pattern may make the most sense—a single instance of each classification marking reused across models*
- Stereotypes also offer significant usage flexibility and facilitate visual cueing through customized formatting
  - *Stereotypes are not inherited—this can be a positive or a negative*
  - *How do you manage the individual components of the security marking vice the aggregation of them?*

Significant tradeoffs between these different implementation alternatives; tool support may also be crucial

Security Classification Guidance Current State

- Current security classification guidance is not DE-ready
  - *Captured in documents that are often not readily ingestible by automated tools*
  - *Captured in natural language without much formal structure*
  - *Terms often not consistently used relative to other SCGs*
  - *Interpretation and application of the marking rules is reliant on human judgment*
    - Time consuming, error-prone, and almost guaranteed to be inconsistently interpreted and applied
  - *Taxonomy of classification rules is not well-defined*

• Term X	• Status or condition of X
• Artifact X	• Method or procedure of X
• Fact of X	• Capability of Y to do X
• Value of X	• Limitations of Y to do X
• Date of X	• Vulnerability of X to Y
• Existence of X	• Details of X
• Rationale of X	• Association of X with Y

Where do we go from here?

Security Classification Guidance Needs to Evolve

- Security classification guidance needs to become more disciplined and DE-ready
  - *Captured in forms that are more readily ingestible by automated tools*
    - Each statement should be able to stand alone; tables or complex sentence structures complicate ingest
  - *Captured with structure—ideally patterns—that facilitate machine understanding*
    - Define and standardize structured patterns of common security classification statements
  - *Terms should be standardized and applied across programs and Services*
    - Security classification guidance and the descriptive models need to use the same terms with the same meaning
  - *Interpretation and application of the marking rules needs to be amenable to automation*
    - Rules not written tightly enough to be repeatable by machines won't be repeatable by humans either
    - Speed, consistency, and confidence are needed to support the tenets of DE

DE implementation is hobbled by 20th century security classification paradigms

Conclusion and Way Forward

- This is a substantially more complex subject than it may appear to be at first glance
  - *Many questions are raised, but there are few if any solutions that are obvious, easy, and defensibly correct*
  - *Existing guidance does not discuss (or even acknowledge) descriptive models as an artifact subject to security classification marking, nor does it address the unique characteristics of models as an artifact*
  - *As a result, interpretation is left as an exercise for the reader*
    - And how sporty do we want to get when it comes to security classification?
- Suggested way forward:
  - *Security classification guidance needs to become more disciplined and DE-ready*
  - *Identify key assertions that we feel comfortable making with explicit classification markings*
    - Straightforwardness measure: Views > Elements > Relationships > Properties
  - *Identify key assumptions that we may need to make even if there is no explicit syntax to represent them in the model*
    - E.g., All sub-elements are assumed to have the same classification as its parent unless otherwise indicated
    - E.g., Mark the value property name even if it's really the/some value of the value property that is classified
  - *Explicitly define the semantics (interpretation) of these assertions for clarity*
  - *Continue to work to identify solutions to address remaining gaps*
  - *Reach consensus on a standard ontology of terms and concepts and a standard set of classification rule patterns*

Standardize across the DoD—this is everyone's problem and we're all operating at risk until we solve it

Questions?

Ryan Noguchi
Principal Engineer
The Aerospace Corporation
ryan.a.noguchi@aero.org