



DoDAF 2.0 Meta Model (DM2)

Overview



**Briefing to the Software Engineering Institute (SEI)
Army Strategic Software Improvement Program
(ASSIP) Action Group (AAG)**

18 Nov 2009

Mr. David McDaniel

Enterprise Architecture & Standards

Office of the DoD Deputy Chief Information Officer



Briefing Outline

- DM2 Purposes
- DM2 Modeling Conventions
- Foundation Ontology
- Partial walkthrough of a sample of DM2 LDM data groups
- Thoughts as to how this could aid software intensive PEO's



Map of DM2 Description Documents and Briefings

Today, you are getting bits of several





The Purpose of DoDAF: Support DoD's core processes

- Capabilities Integration and Development (JCIDS)
- Planning, Programming, Budgeting, and Execution (PPBE)
- Acquisition System (DAS)
- Systems Engineering (SE)
- Operations Planning
- Capabilities Portfolio Management (CPM)



What is the DM2

- Architectural descriptions are models of aspect(s) of the enterprise
 - Common model types are described in DoDAF: AV-1 through DIV-3
 - Other models are called “fit-for-purpose”
 - The DM2 is a model of those models, hence “meta”



DoDAF Meta Model (DM2)

Purposes

1. Establish the ***constrained vocabulary*** for description and discourse about DoDAF models and their usage in DoD's six core processes
2. Specify a way for federated EA ***data exchange*** between:
 - architecture development and analysis tools
 - architecture databases
 - other EA-relevant ADS'
3. Support ***discovery*** and ***understandability*** of EA data:
 - Discovery using DM2 categories of information
 - Understandability via DM2's precise semantics augmented with linguistics (defs and aliases)
4. Increase the ***utility and effectiveness*** of architectural descriptions via a mathematical data model so they can be:
 - Integrated and cross-walked
 - Analyzed, evaluated, and assessed quantitatively



DEPARTMENT OF DEFENSE
CHIEF INFORMATION OFFICER

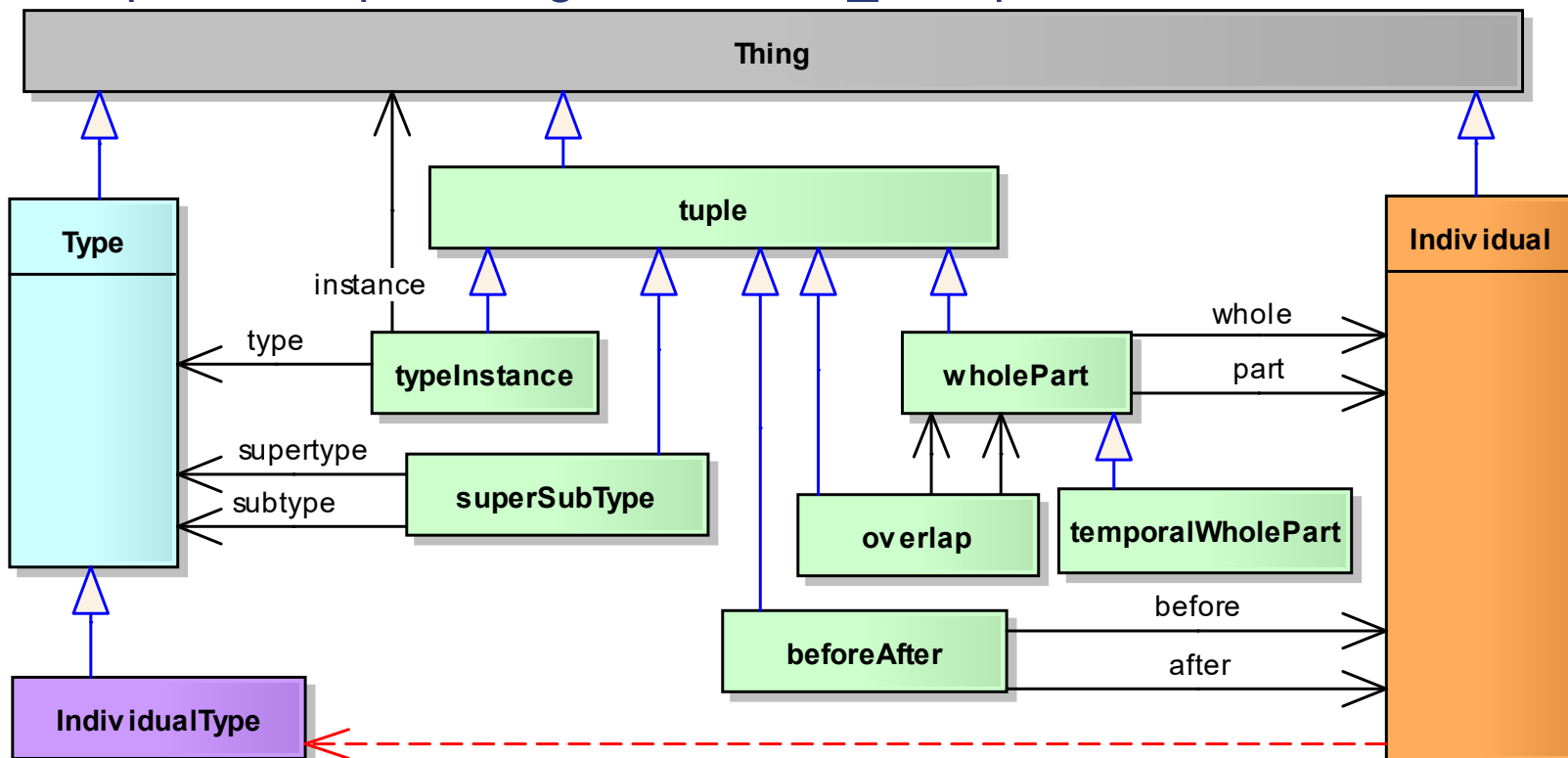
DM2 LDM Conventions



DM2 Development and Maintenance Business Rules

- Scope
- Information Pedigree
- Security classification marking
- Term entry
- Aggregation rule
- Typed Relationships
- Implementation neutrality
- Definitions
- Aliases
- Extensions
- Research and reference information
- DM2 work share site
- Configuration Management
- Pilot and early adopter support

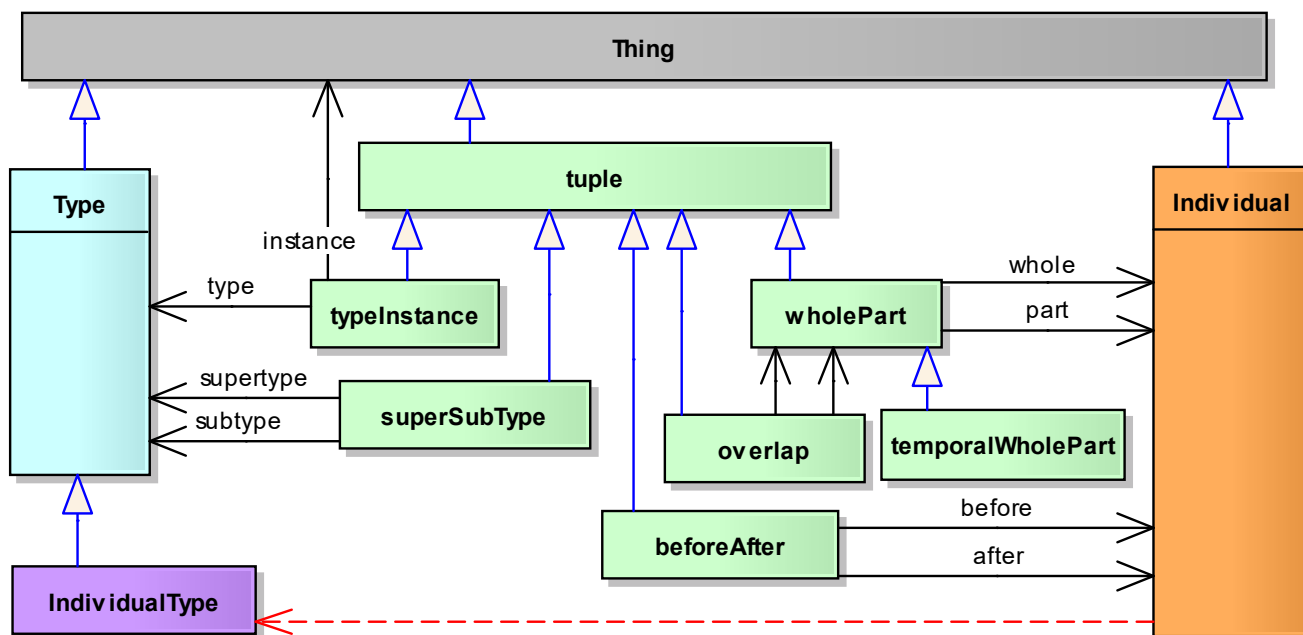
- Developed by an international group of computer scientists, engineers, mathematicians, and philosophers under defense sponsorship
- http://en.wikipedia.org/wiki/IDEAS_Group



Basic Concepts: three types of Things

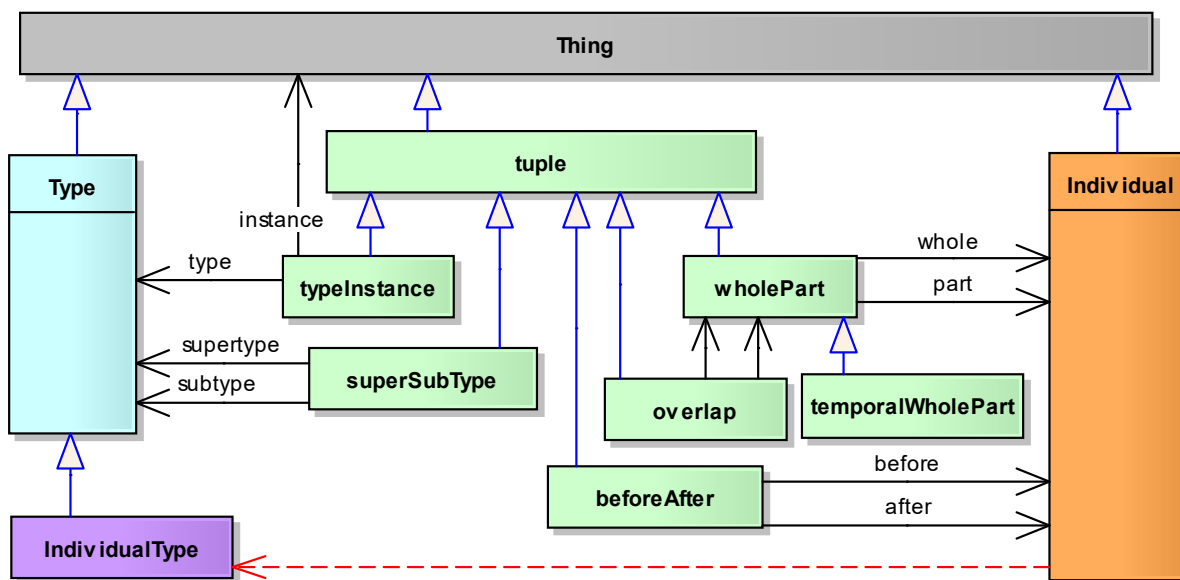


1. Individuals, Things that exist in 3D space and time, i.e., have 4D spatial-temporal extent.
2. Types, similar to sets of Things.
3. Tuples, ordered relations between Things, e.g., ordered pairs in 2D analytic geometry, rows in relational database tables, and subject-verb-object triples in Resource Description Framework.





1. Set theoretic:
 - Super-subtype (categorization of Things); e.g., a type of system or service, capability, materiel, organization, or condition.
 - Type-instance, similar to “element of” in set theory
2. Spatio-temporal parts and wholes (4D mereology):
 - Whole-part; e.g., components of a service or system, parts of the data, materiel parts, subdivisions of an activity, and elements of a measure.
 - Temporal whole-part (temporal states); e.g., the states or phases of a performer, the increments of a capability or projects, the sequence of a process (activity).
3. Spatio-temporal boundaries and overlaps (4D mereotopology):
 - Overlap
 - Before-after (sequences)



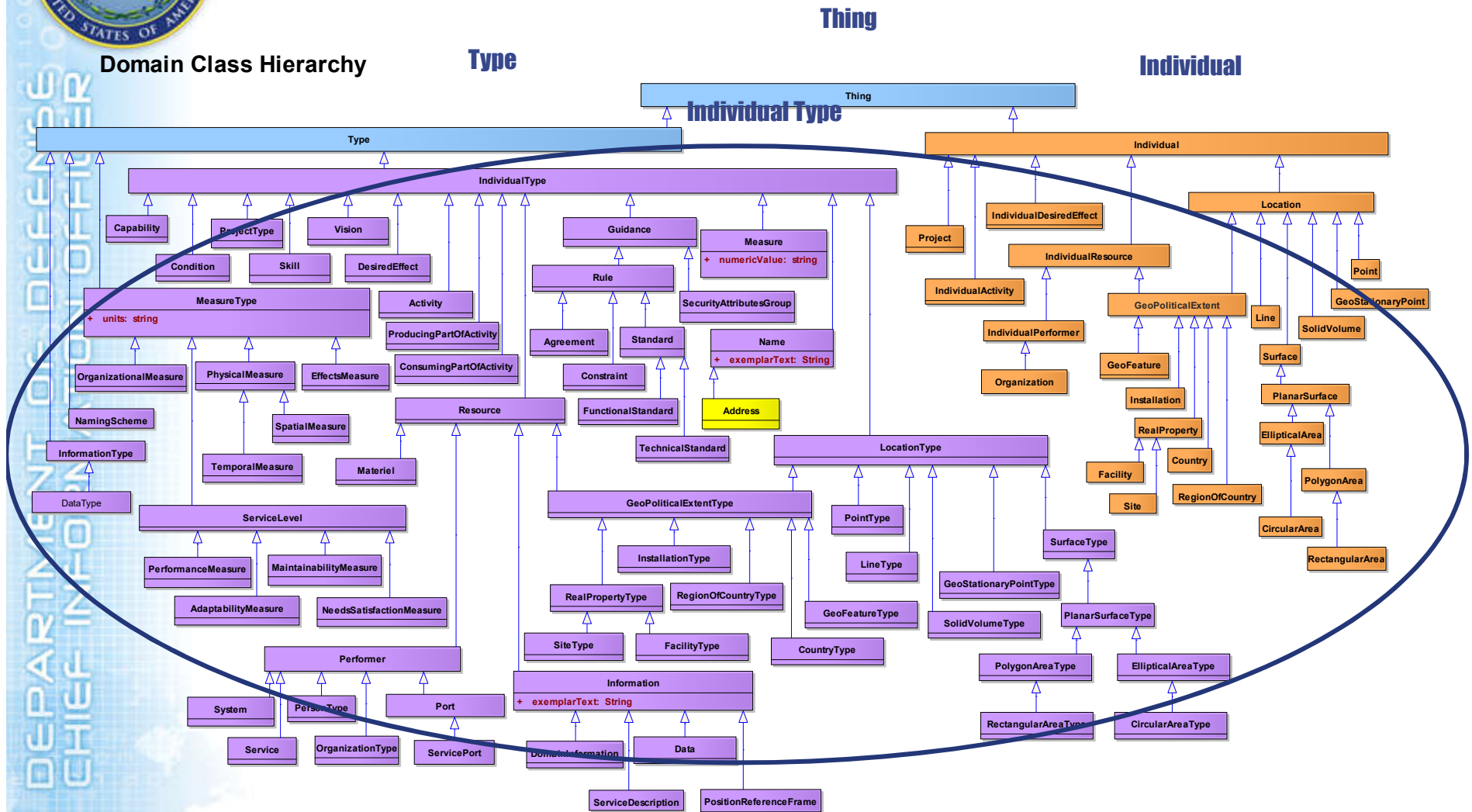
The IDEAS Foundation is:



- Formal, higher-order, 4D, based on **four dimensionalism**
- **Extensional** (see Extension [metaphysics])
 - Using physical existence as its criterion for identity
 - Deals with issues of states, powertypes, measures, space -- **what is truly knowable** vs. what is assumed
 - **signs and representations are separated from referents**
 - The advantage of this methodology is that names are separated from things and so there is no possibility of confusion about what is being discussed.
- Well suited to managing **change-over time**
- **Identifies elements with a degree of precision that is not possible using names alone**
 - Comparing two Individuals, if they occupy precisely the same space at the same time, they are the same.
 - For two Types to be the same, they must have the same members
 - If those members are Individuals, their physical extents can be compared.
 - If the members are Types, then the analysis continues until Individuals are reached, then they can be compared.



DoDAF Domain Concepts are Specializations

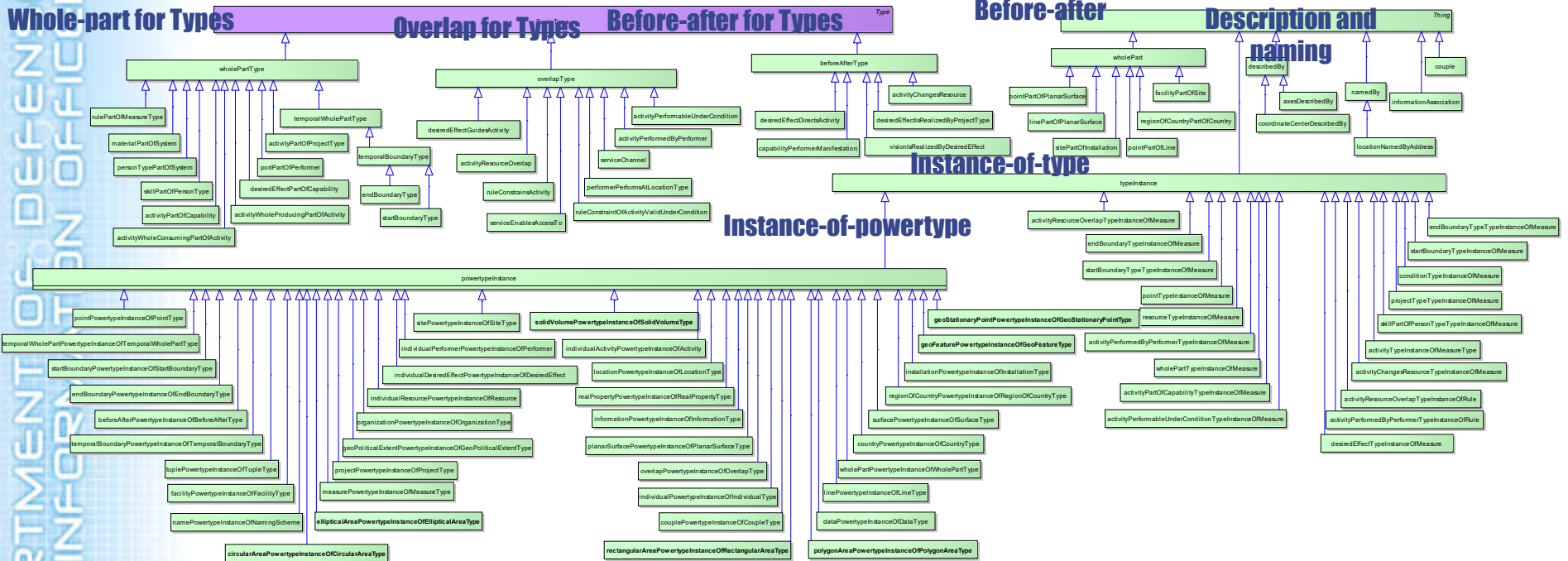


- So they inherit associations (can occupy association place positions)



All Associations are Typed

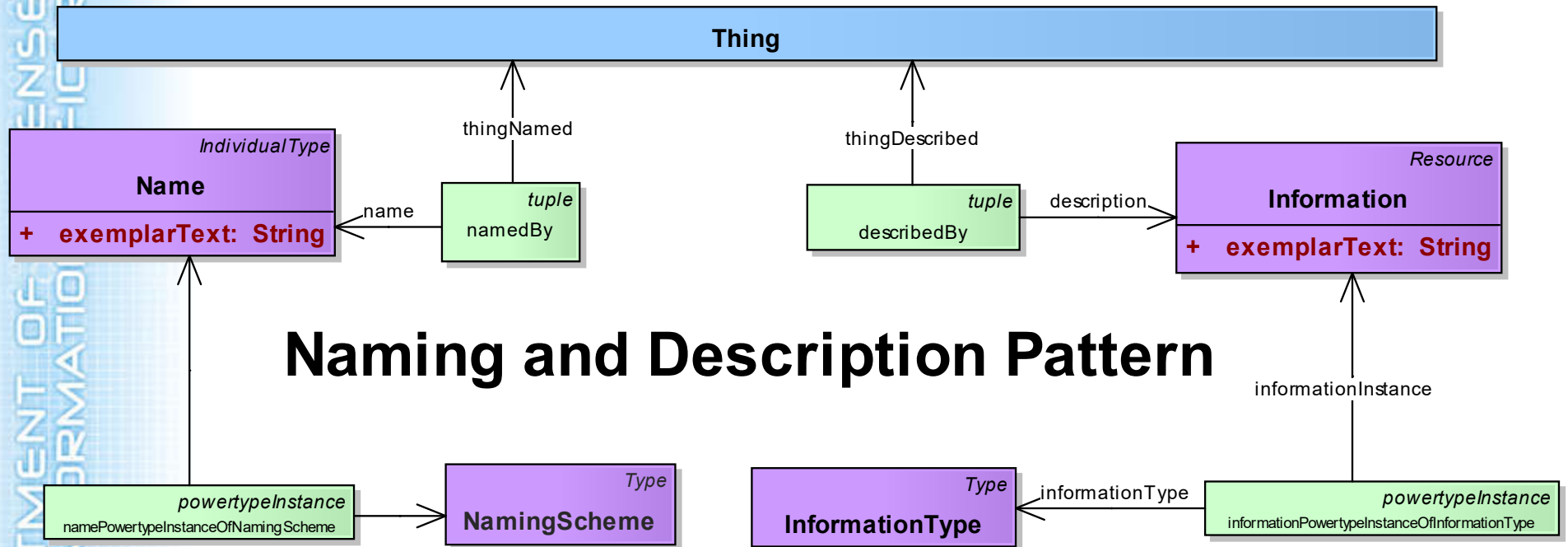
Foundation For Associations



- So their mathematical meaning is formally modeled



Naming and Description Pattern



Naming and Description Pattern

- Multiple names for same thing (aliases) – must tell Naming Scheme
- Information (formerly Information Element) linked to the Things it describes



Benefits of adopting the IDEAS formal foundation and common patterns

- Model compactness through inheritance of superclass properties and rigorously worked-out common patterns.
 - Saved a lot of repetitive work – “ontologic free lunch”
 - Economizes implementations
 - Concentration of effort on a few common patterns results in higher quality and consistency throughout
- Agreed-upon analysis principles that provide a principled basis for issue analysis
- Improved ability to integrate and analyze multiple heterogeneous EA datasets to fulfill EA purposes
 - Depends on near-universal mathematics and science that all learn very similarly



Formal Ontology Mathematics

- Set theory $\in, \supset, \cup, \cap, \dots$
- 4-D (xyzt) mereology (and mereotopology)
 - Whole-part
 - Spatial
 - Temporal
 - Before-after
 - Overlap
- Predicate Calculus $\forall, \exists, \ni, \dots$
- Set theoretic, geometric, and topologic mathematic “laws” and theorems, e.g.,
 - Commutivity and anti-commutivity, e.g., $A \cap B = B \cap A; a \cap b = b \cap a$
 - Symmetry and anti-symetry $aPb \wedge a \neq b \Rightarrow \neg bPa; \quad aBb \Rightarrow \neg bBa$
 - Reflexivity and irreflexivity $A \subset A; \quad aPa; \quad \neg uBu$
 - Associativity
$$A \cup (B \cap C) = (A \cup B) \cap C; \quad A \cap (B \cup C) = (A \cap B) \cup C; \quad aO(bOc) = (aOb)Oc$$
 - Transitivity $A \subset B \wedge B \subset C \Rightarrow A \subset C$
 - Many others $a \in A \wedge A \subset B \Rightarrow a \in B$
if $\{A_i\}$ forms a partition of A then $a \in A_j \Rightarrow a \notin A_k \forall j \neq k$

These types of logics, calculii, theorems, etc. can be applied against datasets founded on formal ontologies such as the IDEAS Foundation



Why math? a spectrum of information sharing

Free-text

Human-interpretable only

Structured document

Human-interpretable but with a predictable organized arrangement

Database

Databases are semantically weaker than commonly understood, e.g., the fundamental concepts of Entity-Relationship and Class Models:



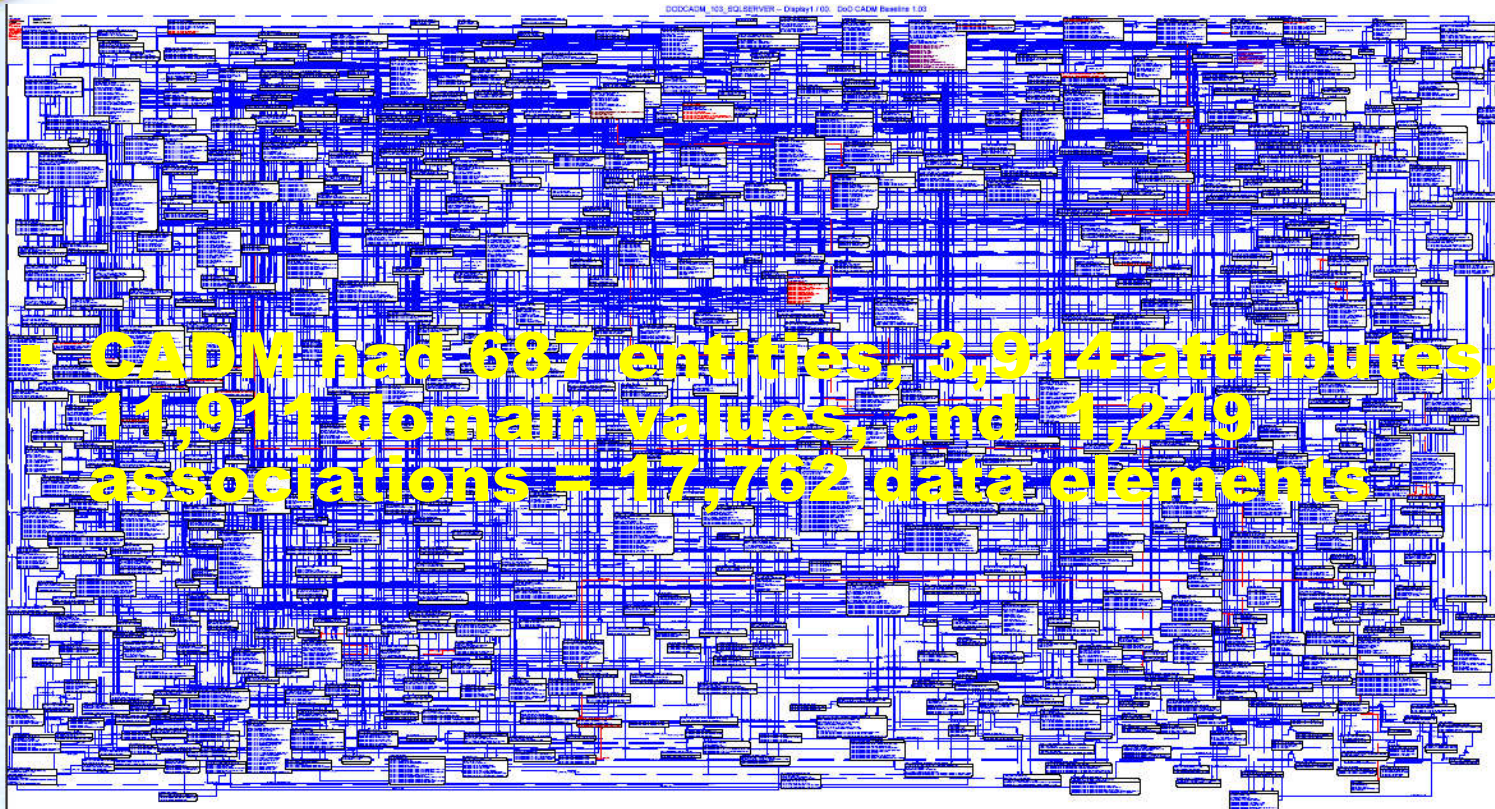
i.e., structured sentences => language-based

Mathematically structured

- Applicable mathematics:
 - Set or type theory
 - Mereology
 - Mereotopology
 - 4 dimensionalism
 - Predicate calculus
 - Logics: modal, Kripke, ...
- Rules, operators:
 - Commutative, reflexive, transitive, ...
 - Member-of, subset-of, part-of, ...



Compare DM2's Predecessor, a classic E-R model



▪ **CADM had 687 entities, 3,914 attributes, 11,911 domain values, and 1,249 associations = 17,762 data elements**

▪ **DM2 has 217 foundation and domain data elements, 37 IC-ISM's, and 4 metadata for a total of 258 data elements**



DM2 LDM Data Groups

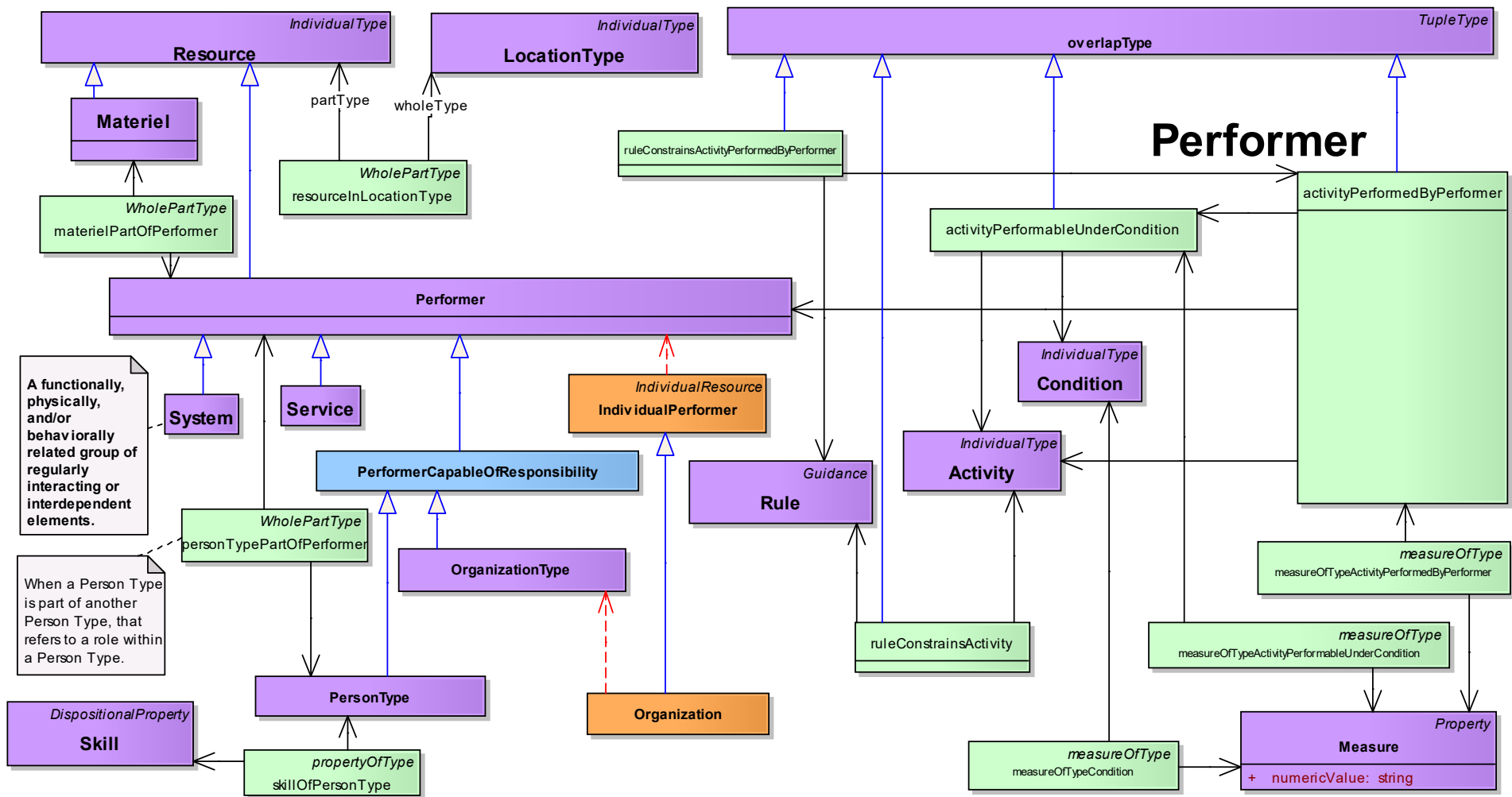
- The DM2 LDM is presented in 12 semantically-related clusters or data groups
 1. Activities
 2. Performer
 3. Resource Flows
 4. Information and Data
 5. Rules
 6. Measures
 7. Locations
 8. Goals
 9. Capability
 10. Services
 11. Project
 12. Training/Skill/Education
- For each group:
 - Diagram, Definitions, and Aliases
 - Notes
 - Method of collection and construction
 - Usage in Core Process

Sample for today's overview



Performer Data Group

- Central to the description of architecture
 - Performers are the “Who”
 - the “How” are assigned to Performers
 - Assignment involves tradeoffs, e.g., for performance and cost/benefit
 - Assignment results in allocated baseline and initial work breakdown structures
- Performers
 - Types: Organizations / Org Types, Person Types, Systems (humans and machines), and Services – or any combination thereof
 - Follow Rules
 - Are at locations
 - Have measures





Performer Data Group Notes

- Performer subtypes:
 - Person Type e.g., Military Occupational Specialties (MOS)
 - MOS describe Skills and their measurement (not shown in this diagram).
 - Person Type includes Materiel assigned and necessary for the performance of activities
 - e.g., as per Army CTA-50.
 - Note that Person Types have temporal whole-parts (states) such as in-garrison or deployed that may have different Materiel compositions and other associations such as applicable Rules.
 - An Organization (type or actual Individual Organization) meaning a mission chartered organization
 - Not limited just collections of people or locations
 - e.g., the Federal Bureau of Investigation (FBI) has a chartered mission and it chooses the locations, people, etc., to accomplish such.
 - A System in the general sense of an assemblage of components – machine, human – that accomplish a function
 - anything from small pieces of equipment to FoS and SoS
 - Systems are made up of Materiel (e.g., equipment, aircraft, and vessels) and Personnel Types, and organizational elements.
 - A Service, from a software service to a business service such as Search and Rescue.
 - Any combination of the above.
- What it means for a Performer to perform an Activity:
 - The performance of an Activity by a Performer occurs in physical space and time.
 - Two ways in which a Performer spatial-temporally overlaps an Activity:
 - In the act of performing the Activity
 - Sometimes called “assigned to” for the purposes of traceability.
 - As part of a larger process (aggregated Activity)
 - Sometimes called “allocated to”
- Not all architecture modeling standards explicitly provide for allocation. For example, the Systems Modeling Language (SysML) extensions to the UML modeling standard have added this feature.

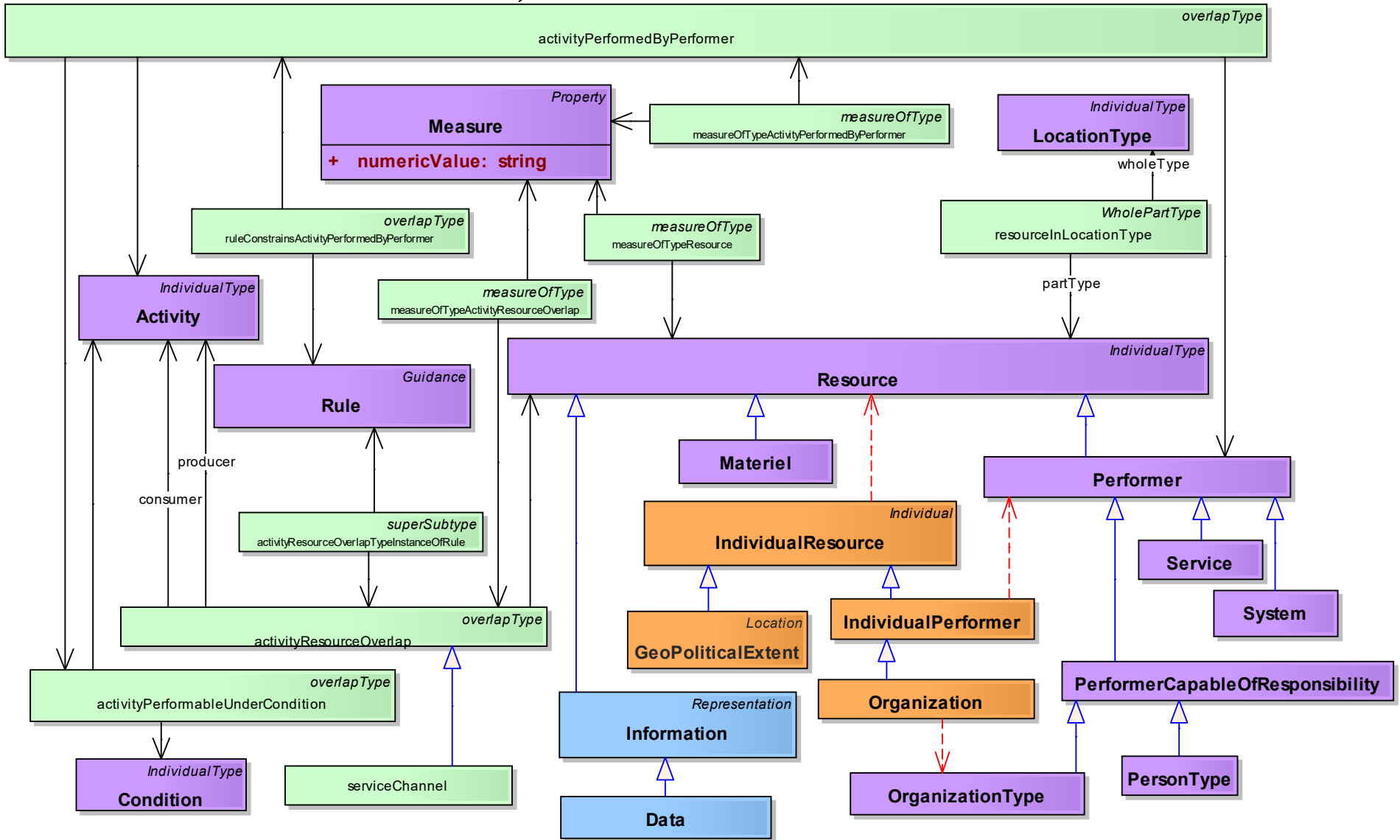


Resource Flow Data Group

- Resource Flows are used to model the flow of Resources – **Materiel**, **Information** (and Data), **Geo-Spatial Extents**, **Performers**, or any combination thereof.
- Resource Flows are key modeling techniques used in the definition of **Interfaces** and assurance of **Interoperability** between Activities and their performing Performers (e.g., Systems and Personnel.)
- Resource Flow models and associated analysis techniques reveal behavior such as:
 - a. The connectivity between resources.
 - b. Resource Flow modeling provides an explicit means to describe the behavior of activities, systems, organizations and their composite effects on the overall enterprise.
 - c. The **content** of the information flowing between resources (e.g., interface definition).
 - d. The **order or sequential behavior** (parallel or serial) of the resources in relation to one another (e.g., project task execution and critical path).
 - e. The behavior of Resource Flow between or within organizations (e.g., work flow, information flow, etc.).
 - f. The changes in state during the spatial and/or temporal existence of the resource.
 - g. The rules that modify the behavior of the Resource Flow (e.g., business rules, controls, decisions, etc.).
 - h. The measures that define the quality, constraints, timing, etc. of the Resource Flow (e.g., Quality of Service (QoS), measures of performance, measures of effectiveness, etc.).
 - i. The **flow of control** orchestrating the behavior of the Resource Flow.

Resource Flow

i.e., Role





Resource Flow Data Group Notes

- The term flow implies that something (e.g., materiel, information) is moving from point A to point B, hence the use of the foundation concept of “overlap”.
- **Resource Flows are Activity-based**, not Performer based since a Performer cannot produce or consume a resource other than by conduct of a production or consumption activity.
- Whereas prior versions of DoDAF modeled only information and data exchanges and flows, this version also allows modeling of other flows, such as:
 - **Materiel flows** such as ammunition, fuel, etc. important for modeling the fire rate, logistics, etc., aspects of a Capability solution so it can be compared with other alternative solutions.
 - **Personnel Types** such as Military Occupational Specialty (MOS) that allow representation of the Training and Education pipeline aspects of Doctrine, Organization, Training, Material, Leadership and Education, Personnel, and Facilities (DOTMLPF).
 - **Performers** such as Services, Systems, or Organizations that might be the output or result of a Project’s design and production process (activities). This allows modeling of, for instance, an acquisition project.
- The exchange or flow triple may have standards (Rules) associated with it such as Information Assurance (IA)/Security rules or, for data publication or subscription, data COI and web services standards.
- The exchange or flow triple may have Measures associated with it such as timeliness, throughput, reliability, or QoS.
- Resource Flow modeling can be performed at varying levels of detail and fidelity depending on the areas of concern being analyzed and the solutions being sought. The upper-level aggregations have been termed *need lines* in previous versions DoDAF. Other terminology expressing levels of aggregation are used depending on the community of interest (e.g., The SysML modeling standard uses *lifeline*).



Capability Data Group

- The Capability Data Group provides information on the collection and integration of activities that combine to respond to a specific requirement.
- A capability, as defined here is *“the ability to achieve a **desired effect** under specified standards and conditions through combinations of **means and ways** to perform a set of **tasks**.”*
- This definition is consistent with that contained in the JCIDS Instruction published by the Joint Staff



Capability Data Group Notes

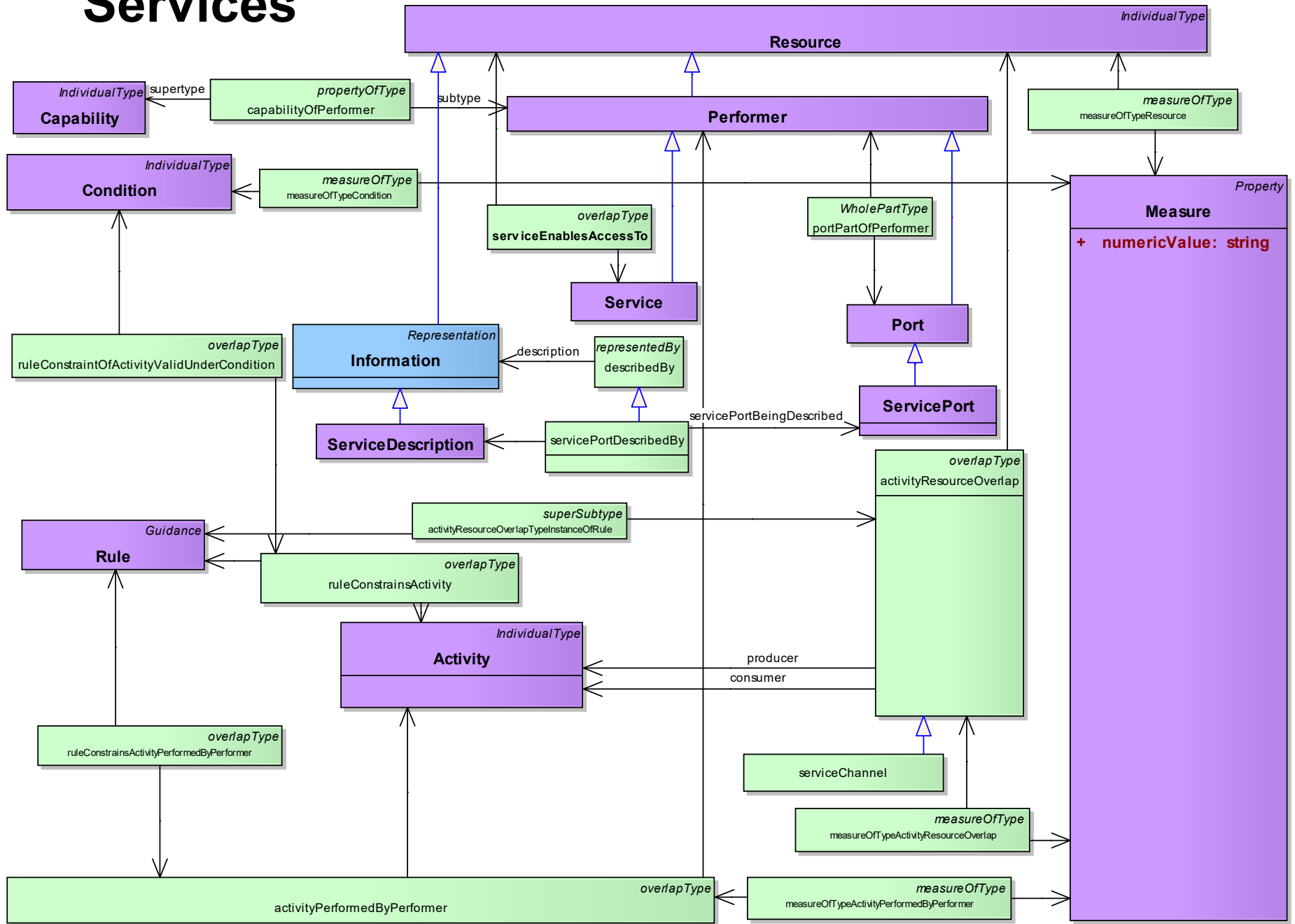
- “Ways and means” are interpreted in DM2 language to be **Activities and Resources**
- Because a **Desired Effect** is a desired state of a Resource (see Goals data group), a Capability is about states (persistence of current ones, or changes to future ones) of Resources.
- Capabilities link to **Measures** (Metrics) through the **Activities** they entail and the **Desired Effects** sought. (see next slide)
- Capabilities relate to Services via the realization of the Capability by a Performer that is a Service. In general, a Service would not provide the Desired Effect(s) but, rather, access to ways and means (Activities and Resources) that would.



Services Data Group

- A Service, in its broadest sense, is a well-defined way to provide a unit of work, through which a provider provides a useful result to a consumer.
 - Services do not necessarily equate to web-based technology or functions, although their use in the net-centric environment generally involves the use of web-based, or network-based, resources.
- Services form a pool that can be orchestrated in support of operational activities, and the operational activities define the level of quality at which the Services are offered.

Services





Services Data Group Notes

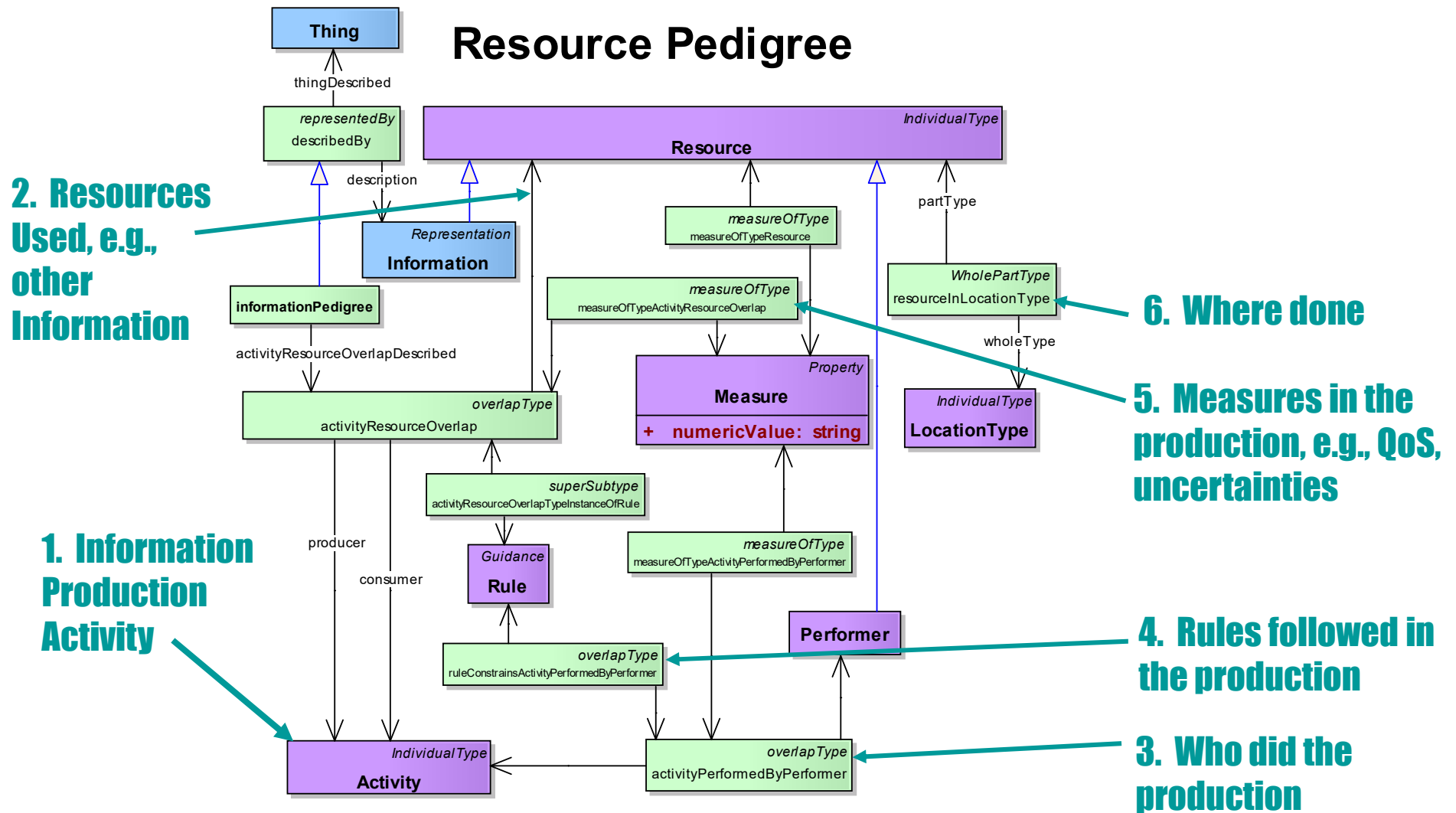
- A Service is a type of Performer which means that it **executes an activity and provides a capability**.
- Unlike DoDAF V1.5, Services in DoDAF V2.0 **include business services**, such as Search and Rescue. This is important to keep in mind because much of the SOA literature is IT-oriented.
- Capabilities and Services are related in two ways. One, the realization or implementation of a Capability by a Performer (usually a configuration of Performers, including Locations) may include within the configuration Services (or Service compositions) to access other Performers within the overall Performer configuration. Conversely, the realization or implementation of a Capability by a Performer (configuration, including Location) may provide the Performers that are accessed by a Service (or Service composition).
- Since Service inherits whole-part, temporal whole-part (and with it before-after), **Service may refer to an orchestrated or choreographed Service**, as well as individual Service components.
- Any Performer that consumes a Service may have a **Service Port** that is described in the service request.



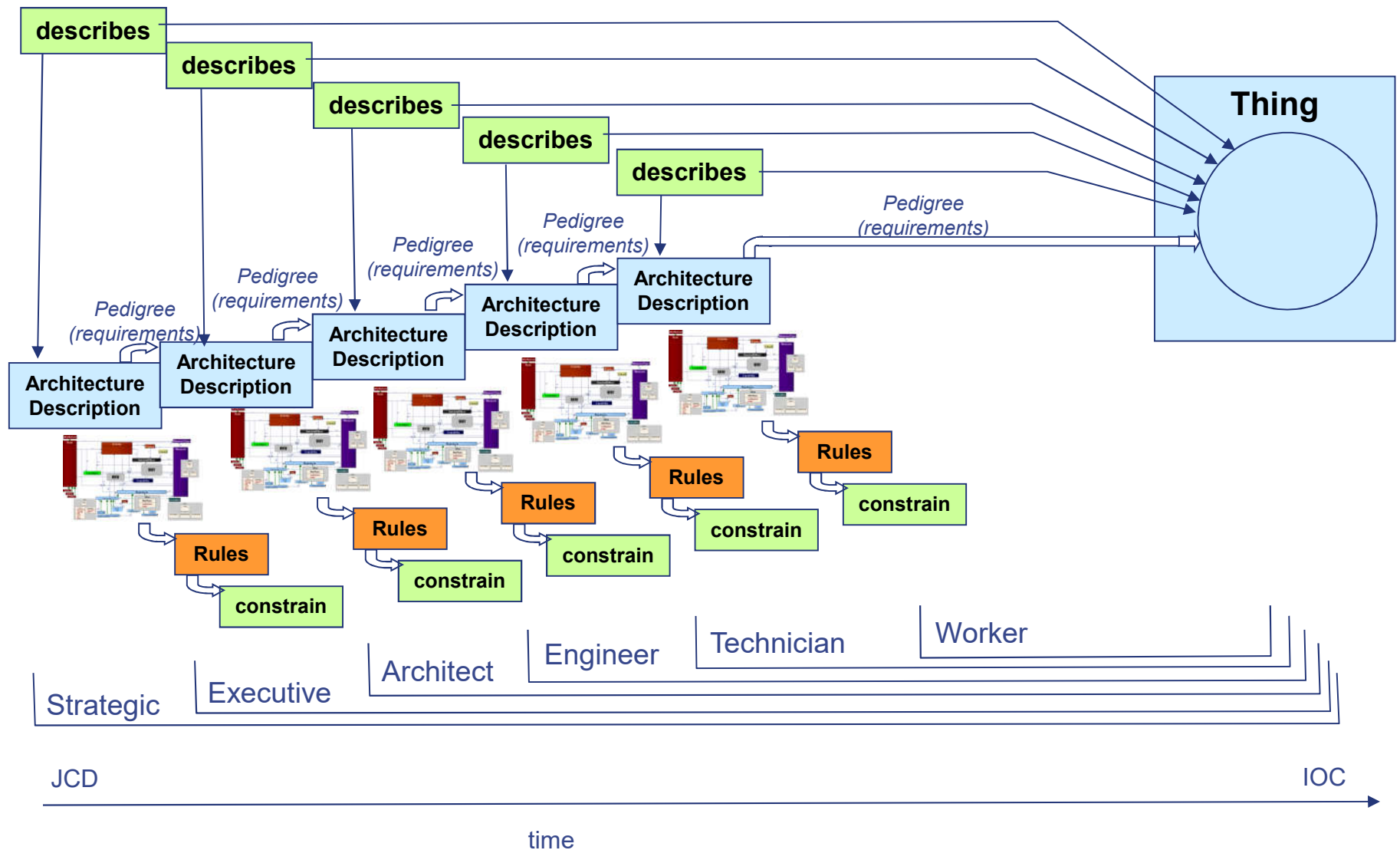
Services Data Group Notes

- A Service Port is a special type of Port that is
 - Self-describing and visible. The Service Description of the Service Port is of all aspects necessary to utilize the Service and no more.
 - As such, it may include visible functionality, QoS, interface descriptions, data descriptions, references to Standards or other Rules (Service Policy), etc.
 - The inner workings of the Service are not described in a Service Description.
 - Part of a Performer that provides access to the Performer capabilities and the consumer that participates in the service.
 - Note that the Performer capabilities provided access to can be an aggregate, e.g., an orchestration, of Performer components.
 - The Service Port is the service consumer facing part of the Service and so has a Service Description, a type of Information.
 - Since Service Ports are types of Ports and Ports are types of Performers, they inherit all of Performer's properties, including Measures associated with the Performer, performance of Activities (Service Functions) with associated Measures, and provision of objects (Materiel, Data, Information, Performers, Geopolitical Extents).

Information Pedigree – workflow model



Requirements Traceability and Design Levels in DM2





DM2 Possible Interest for Software Intensive PEOs: (1) As it makes DoDAF work better

- SoS Engineering and Integration
 - Upgrade orchestration – no more “ATO in the Gulf”
 - Interoperability
 - Specification for – comms, data, processing, procedures, ...
 - Assessment during acquisition
 - Solutions that fit Capability objectives
 - Traceability through design refinement
- Data Strategy / Data Sharing
 - Information sharing requirements modeling
 - COI DMWG’s and interactions, ADS strategies,
 - Derivation of data schemas from information requirements -> conceptual models -> logical -> physical schema
- Network consolidation
 - Connectivity requirements – rules, metrics, business processes, ...



DM2 Possible Interest for Software Intensive PEOs: (2) the mathematical foundation itself

- The mathematical foundation itself, as an input to new ways for PEO's to architect software data structures supporting, e.g.,
 - Componentization
 - Interoperability of independently acquired systems and components
 - Integration scaling
 - SoA orchestration
 - Unanticipated data use



DEPARTMENT OF DEFENSE
CHIEF INFORMATION OFFICER

Questions?

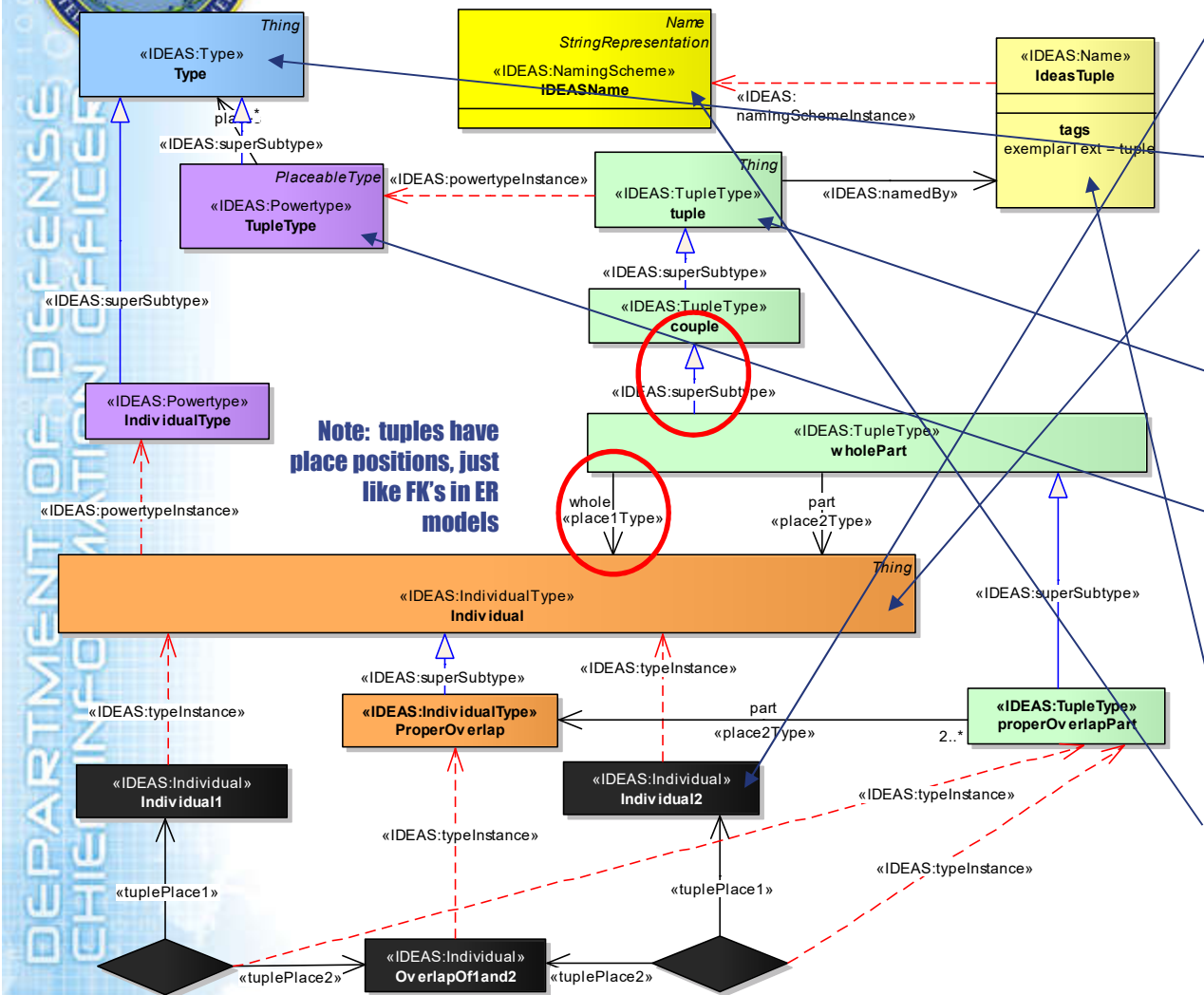


DEPARTMENT OF DEFENSE
CHIEF INFORMATION OFFICER

Backup



LDM Diagramming and Use of UML as an Ontology Diagramming Tool



Individual -- An instance of an Individual - something with spatio-temporal extent

Type -- The specification of a Type

IndividualType -- The specification of a Type whose members are Individuals

TupleType -- The specification of a Type whose members are tuples

Powertype -- The specification of a Type that is the set of all subsets of a given Type

Name -- The specification of a name, with the exemplar text provided as a tagged value

NamingScheme -- The specification of a Type whose members are names



Some points about the foundation:

- Types include sets of Tuples and sets of sets.
- Tuples can have other Tuples in their tuple places.
- There are three subtypes of Type: 1) Individual Type, sets whose members are Individuals (Things with spatio-temporal extent); Power Types, sets whose members are generated from a powerset on some other set; and 3) Tuples, sets of ordered relations between Things.
- The participants in a super-subtype relationship can be from the same class, e.g., the supertype can be an instance of Measure Type as well as the subtype. This allows for representation of as much of a super-subtype taxonomy as is needed.



Powertypes

- Power Type members are generated from some Type by taking all the possible subsets of the members of the Type. For example consider the Type whose members are a, b, c. The powerset would be:

$$\{a,b,c\}, \{a,b\}, \{a,c\}, \{b,c\}, \{a\}, \{b\}, \{c\}, \{\emptyset\}$$

- Some of these subsets are not used by anyone, e.g., the full set, the null set, or just some random subset.



Interesting Instances of Powertypes

- Take the Individual Type AIRCRAFT, whose members include all the aircraft of the world. The powerset generated from this set would have:

$$\{a_1, a_2, \dots, a_n\}, \{\emptyset\}$$

$$\{F-15_1, F-15_2, \dots, F-15_{lastF-15built}\}$$

$$\{F-15_1, 747_1, \dots, Cessna_1\}$$

- The first two are not very interesting
- The second one, which might be name F-15 Type, is quite useful.
- The last example is not useful to most unless you are interested in the first (assuming the subscript 1 means first) of any particular aircraft type, e.g., if you were doing a study of first-off-the-line aircraft production lessons-learned.
- The usefulness of Power Types
 - they allow for multiple categorization schemes with traceability back to the common elements so that the relationships between multiple categorization schemes are known
 - multiple categorization schemes or taxonomies in EA because across a large enterprise it is not possible to employ a single categorization scheme, rather schemes vary depending on function.
 - For example, a weaponeer's classifies ordnance is naturally different from a logistician's, the former concerned with delivery means, lethality, etc. and the latter with weight, size, and other transportation issues.
- Note also that a powerset can then be taken of the powerset