DoDAF 2.0 Meta Model (DM2)
Ontologic Foundation and Pedigree Model

NCSC

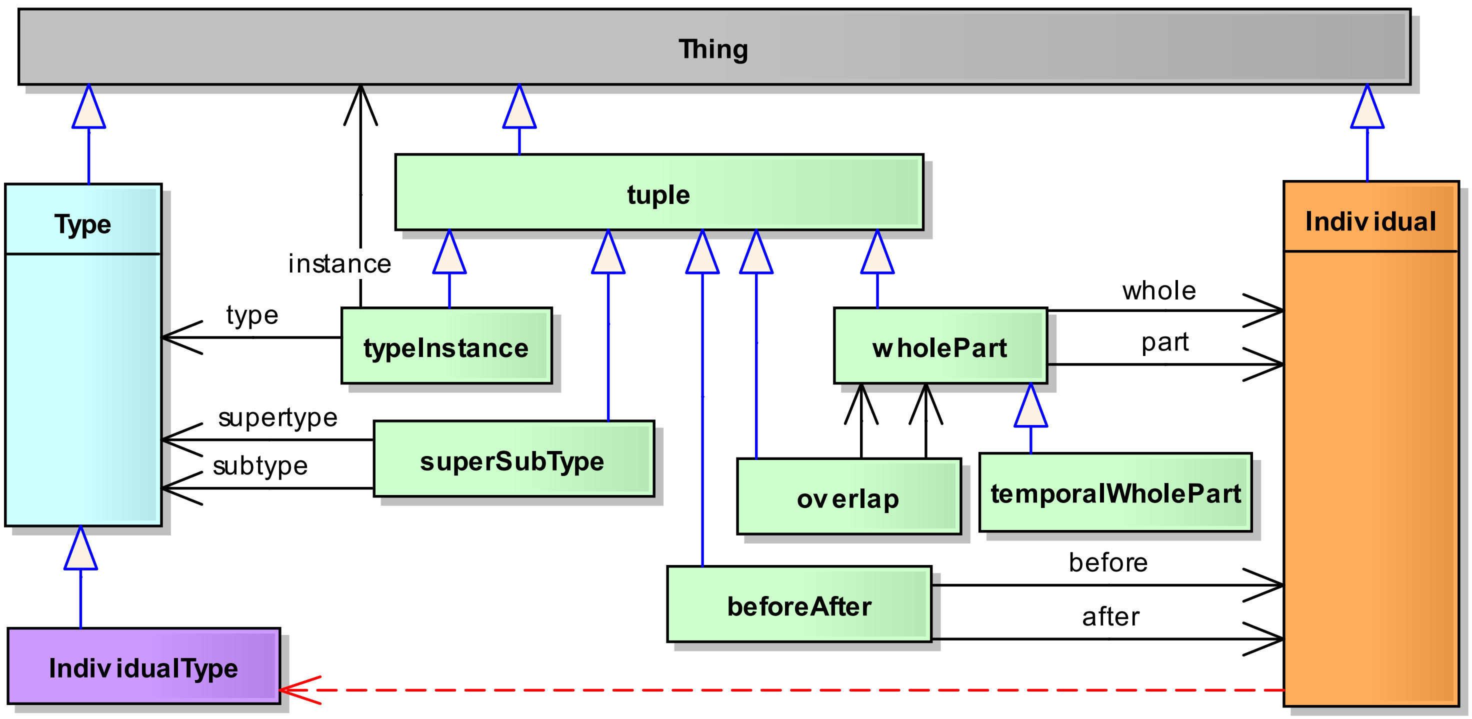
08 October 2009

Briefing Outline

- THE DM2 FOUNDATION
- DODAF PHYSICAL EXCHANGE SPECIFICATION
- EXCHANGE OF DM2 PES XML DOCUMENTS
- PES XSD XML DOCUMENT EXAMPLES
  - UPDM SEARCH AND RESCUE
  - ISP SAMPLES

IDEAS Foundation Top-Level

- The DM2 is ontologically founded upon the International Defence Enterprise Architecture Specification (IDEAS), from which all DoDAF concepts inherit many important properties



The IDEAS Foundation is:

- Formal, higher-order, 4D, based on four dimensionalism
- Extensional (see Extension [metaphysics])
  - using physical existence as its criterion for identity
  - well suited to managing change-over time and identifying 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.
  - Deals with issues of states, powertypes, measures, space -- what is truly knowable vs. what is assumed
- 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.
- Separates signs and representations 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.

Basic Concepts

- Three basic types of Things:
  1. Individuals, things that exist in 3D space and time, i.e., have 4D spatial-temporal extent.
  2. Types, 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.
- Basic relationships:
  - Set theoretic:
    - Super-subtype; e.g., a type of system or service, capability, materiel, organization, or condition.
    - Type-instance, similar to “element of” in set theory
  - Mereologic:
    - 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; e.g., the states or phases of a performer, the increments of a capability or projects, the sequence of a process (activity).
  - 4D Mereotopology:
    - Overlap
    - Before-after

Why Formal Ontology?

- Mathematical rigor needed for precision Architectural Descriptions that can be analyzed and used in detailed processes such as Systems Engineering and Operations Planning.
  - Better ability to integrate and analyze EA data for EA purposes.
- DM2 domain concepts are extensions to the formal foundation
  - Rigorously worked-out common patterns are reused: Super-subtype, whole-part, temporal whole-part, type-instance, before-after, overlap
  - Saved a lot of repetitive work – "ontologic free lunch"
  - Model compactness through inheritance of superclass properties and common patterns.
  - Economizes implementations
  - Result is higher quality and consistency throughout
- Improved interoperation with Unified Profile for DoDAF and MODAF (UPDM)-SysML tools which are following IDEAS concepts.
- Improved opportunities for Coalition and NATO data exchange since MODAF is following IDEAS and NAF is interested in following IDEAS.
- Agreed-upon analysis principles that provide a principled basis for issue analysis

Benefits of Rigorously Structured EA Data

A spectrum of information sharing:

Free-text Human-interpretable only

Structured document Human-interpretable but with a predictable organized arrangement

Database

Normally little more semantic structure than structured text

- Named records (or tables or classes) that are some sort of container for named fields (or attributes or columns).
- Associations and relationships, containers can point to information in other containers
- Because of the labeling, you can tie the information together and query them. A SQL query is just fundamentally a selection of the information.
- Referential integrity, data validation, cardinality rules, etc.

Mathematically structured

- Applicable mathematics:
  - Set or type theory
  - Mereology
  - Mereotopology
  - 4 dimensionalism
  - Predicate calculus
  - Logics: modal, Kripe, …
- Rules, operators:
  - Communtivity, reflexivity, transitiviy, …
  - Member-of, subset-of, part-of, …

Benefits of Rigorously Structured EA Data

- Databases are really just storage and retrieval with connections only for exactly matching pieces of information (e.g., "keys" or exactly matching strings).
- The nature and purposes of EA require more than just storage, retrieval, and exchange, e.g., integration, analysis, and assessment across datasets
- For example, the logical entailment of an EA dataset or collection of related EA datasets might reveal inconsistencies.
- EA entailment examples:
  - "F-16's can fly at least Mach y" ==> F-16C's can fly at least Mach y
  - "Ship's Self Defense System can parse and generate TADIL-J messages" and "SSDS is-part-of all CVNs" ==> CVN's can parse and generate TADIL-J messages
- Without the "intelligence" to perform entailment, data integrations, queries, and analysis algorithms miss connections.
- DM2's ontologic foundation supports entailment and other sorts of mathematical understanding of the data with membership (~ set theory) and 4D mereotopology (parts and boundaries).
  - These are so fundamental in human reasoning that it's hard to see that computers don't have it at all
  - Needed if we want to use them for something more than just storage and retrieval.
  - Has to be encoded it into them with formal methods

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:

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

- 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

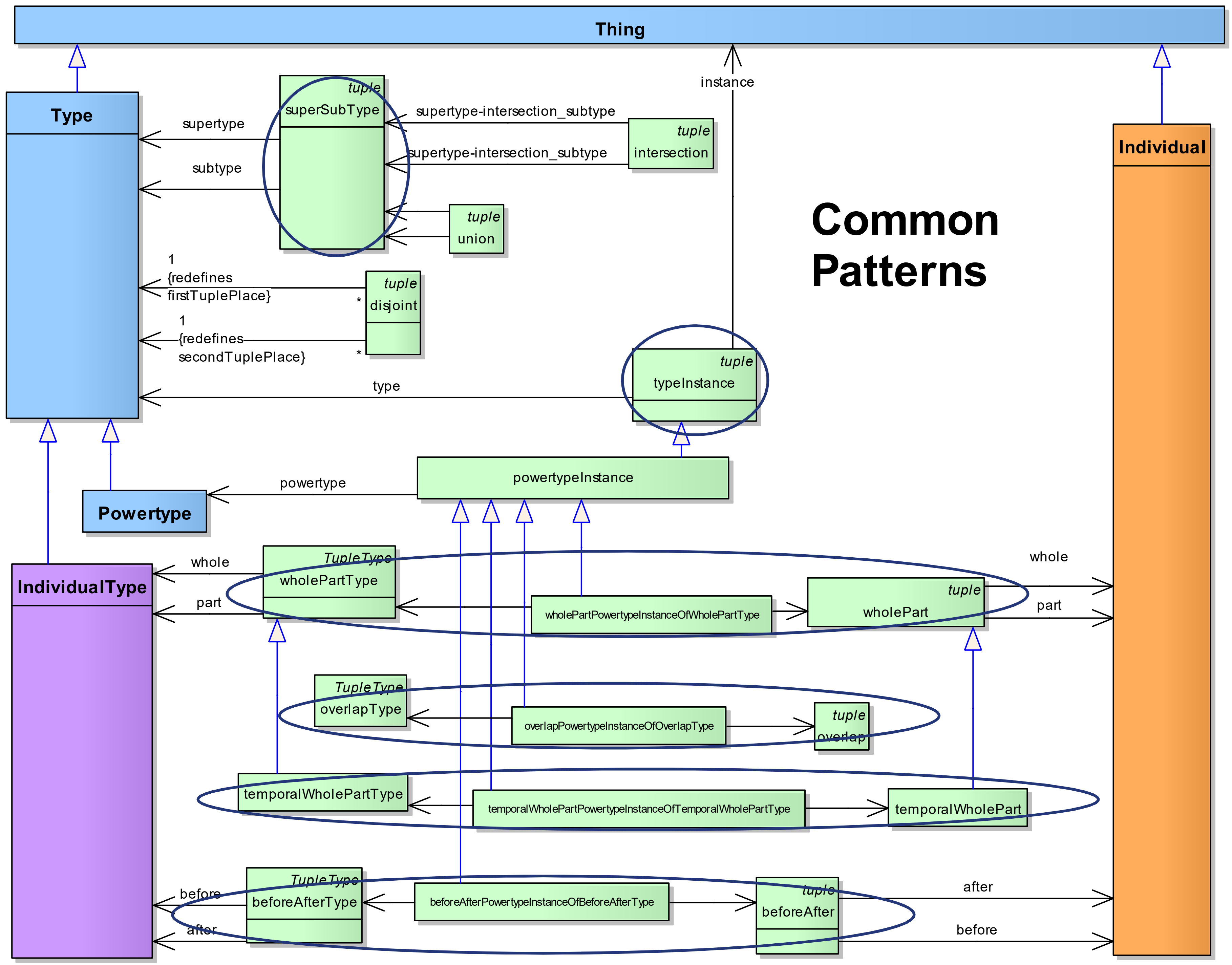
- Take the Individual Type AIRCRAFT, whose members include all the aircraft of the world. The powerset generated from this set would have:

$$\{a_1, a_2, ..., a_n\}, \{\varnothing\}$$

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

$$\{\text{F-15}_1, 747_1, ..., \text{Cessna}_1\}$$

  - 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



Thing
instance
Type
Individual
Common Patterns
tuple
superSubType
supertype
subtype
supertype-intersection_subtype
supertype-intersection_subtype
tuple
intersection
tuple
union
1
{redefines firstTuplePlace}
*
tuple
disjoint
1
{redefines secondTuplePlace}
*
tuple
typeInstance
type
powertypeInstance
powertype
Powertype
IndividualType
whole
part
TupleType
wholePartType
wholePartPowertypeInstanceOfWholePartType
tuple
wholePart
whole
part
TupleType
overlapType
overlapPowertypeInstanceOfOverlapType
tuple
overlap
temporalWholePartType
temporalWholePartPowertypeInstanceOfTemporalWholePartType
temporalWholePart
before
after
TupleType
beforeAfterType
beforeAfterPowertypeInstanceOfBeforeAfterType
tuple
beforeAfter
after
before

IDEAS Foundation Concepts

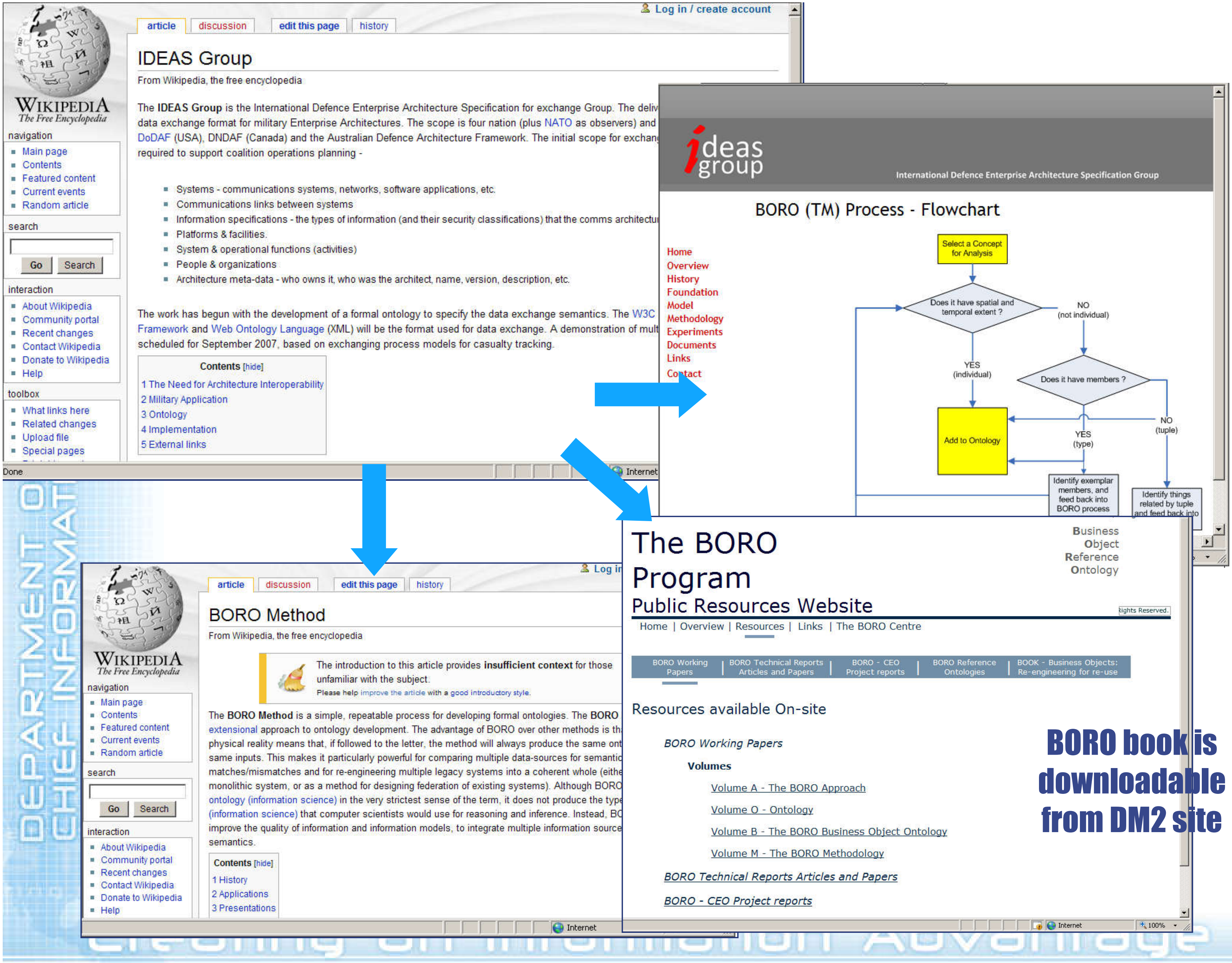
IDEAS Concept	Definition
Classes	
endBoundary	The maximum time value of a temporal extent.
endBoundaryType	The maximum value of a temporal extent taken over a Type, i.e., the maximum time value taken over all it's members.
Individual	A Thing that has spatio-temporal extent. Note - this may be something that existed in the past, exists now, or may exist in some future possible world.
IndividualType	The powertype of Individual.
Information	Information is the state of a something of interest that is materialized -- in any medium or form -- and communicated or received.
InformationType	Category or type of information
Name	The type of all utterances of a given name for a Thing. The exemplarText provides a written example of the uttered name.
NamingScheme	A Type whose members are Names. What kind of name the name is.
Powertype	A Type that is the set (i.e., Type) of all subsets (i.e., subTypes) that can be taken over the some Type.
startBoundary	The beginning of a temporalBoundary.
temporalBoundary	The start and end times for the spatio-temporal extent of an Individual
temporalBoundaryType	The start and end times for the Individual members of a Type.
Thing	The union of Individual, Type, and tuple.
TupleType	The powertype of tuple that provides the stereotype for tuples of Types.
Type	A set (or class) of Things. Note1: Types are identified by their members (i.e. all the things of that type). Note2: The IDEAS Foundation is a higher-order ontology, so Types may have members that are also Types.

IDEAS Foundation Associations

Associations	
beforeAfter	A couple that represents that the temporal extent end time for the individual in place 1 is less than temporal extent start time for the individual in place 2.
beforeAfterPowertypeInstanceOfBeforeAfterType	beforeAfter is a member of BeforeAfterType
beforeAfterType	An association between two Individual Types signifying that the temporal end of all the Individuals of one Individual Type is before the temporal start of all the Individuals of the other Individual Type.
couple	An ordered relationship (tuple) between two Things, i.e., that has two place positions.
couplePowertypeInstanceOfCoupleType	couple is a member of CoupleType
coupleType	A couple in which the places are taken by Types only.
describedBy	A tuple that asserts that Information describes a Thing.
disjoint	Asserts that two Types define disjoint sets (i.e. they share no common members).
endBoundaryPowertypeInstanceOfEndBoundaryType	endBoundary is a member of EndBoundaryType
endBoundaryTypeInstanceOfMeasure	endBoundary is a member of Measure
endBoundaryTypeTypeInstanceOfMeasure	endBoundaryType is a member of Measure
individualPowertypeInstanceOfIndividualType	individual is a member of IndividualType
informationPowertypeInstanceOfInformationType	information is a member of InformationType
intersection	A couple of couples where each constituent couple represents the subset that is common to both sets.
namedBy	A couple that asserts that a Name describes a Thing.
namePowertypeInstanceOfNamingScheme	Name is a member of NameType
overlap	A couple of wholePart couples where the part in each couple is the same.
overlapPowertypeInstanceOfOverlapType	overlap is a member of OverlapType

IDEAS Foundation Associations

overlapType	An overlap in which the places are taken by Types only.
powertypeInstance	An association that between of the sets within the powerType and the powerType. A special form of typeInstance.
startBoundaryPowertypeInstanceOfStartBoundaryType	startBoundary is a member of startBoundaryType
startBoundaryType	The beginning of a temporalBoundaryType.
startBoundaryTypeInstanceOfMeasure	startBoundary is a member of Measure
startBoundaryTypeTypeInstanceOfMeasure	startBoundaryType is a member of Measure
superSubType	An association in which one Type (the subtype) is a subset of the other Type (supertype).
temporalBoundaryPowertypeInstanceOfTemporalBoundaryType	temporalBoundary is a member of temporalBoundaryType
temporalWholePart	A wholePart that asserts the spatial extent of the (whole) individual is co-extensive with the spatial extent of the (part) individual for a particular period of time.
temporalWholePartPowertypeInstanceOfTemporalWholePartType	temporalWholePart is a member of temporalWholePartType
temporalWholePartType	A couple between two Individual Types where for each member of the whole set, there is a corresponding member of the part set for which a wholePart relationship exists, and conversely
tuple	A relationship between two or more things. Note: Tuples are identified by their places (i.e. the ends of the relationship).
tuplePowertypeInstanceOfTupleType	tuple is a member of TupleType
typeInstance	A Thing can be an instance of a Type - i.e. set membership. Note that IDEAS is a higher-order model, hence Types may be instances of Types.
union	A couple of couples where each constituent couple represents the superset union over the unioned sets.
wholePart	A couple that asserts one (part) Individual is part of another (whole) Individual.
wholePartPowertypeInstanceOfWholePartType	wholePart is a member of wholePartType
wholePartType	A coupleType that asserts one Type (the part) has members that have a whole-part relation with a member of the other Type (whole).



BORO book is downloadable from DM2 site

Initial work on mathematics of data modeling

- Set theory $\in, \subset, \cup, \cap, \ldots$
- 4-D (xyzt) mereology (and mereotopology)
  - Whole-part
    - Spatial
    - Temporal
  - Before-after
  - Overlap
- Predicate Calculus $\forall, \exists, \ni, \ldots$

Depends on near-universal mathematics and science that all learn very similarly

Examples of some set theoretic formalisms

Commutative and anti-commutative, e.g., $A \cap B = B \cap A$

Reflexive and irreflexive, e.g., A $\subset$ A, A $\subsetneq$ A

Associative, e.g., $A \cup (B \cup C) = (A \cup B) \cup C; \quad A \cap (B \cap C) = (A \cap B) \cap C;$

Transitive, e.g., A $\subset$ B $\wedge$ B $\subset$ C $\Rightarrow$ A $\subset$ C

others:

$a \in A \wedge A \subset B \Rightarrow a \in B$

if $\{A_i\}$ forms a partition of A then $a \in A_j \Rightarrow a \notin A_k \forall j \neq k$

Elements, Subsets, and Powersets

- “is-a” example:

Appears to

Aristotle is-a sapiens is-a species $\Rightarrow$ Aristotle is-a species

- Using mathematical constructs:

$$\text{Aristotle} \in \text{sapiens} \wedge \text{sapiens} \in \text{species} \nRightarrow \text{Aristotle} \in \text{species}$$

- Powersets

$\text{sapiens} \subset \text{homo} \subset \text{hominidae} \subset \text{primate} \subset \text{mammal}$

$\text{sapiens} \in \text{species}$

$\text{homo} \in \text{genus}$

$\text{hominidae} \in \text{family}$

$\text{primate} \in \text{order}$

$\text{mammal} \in \text{class}$

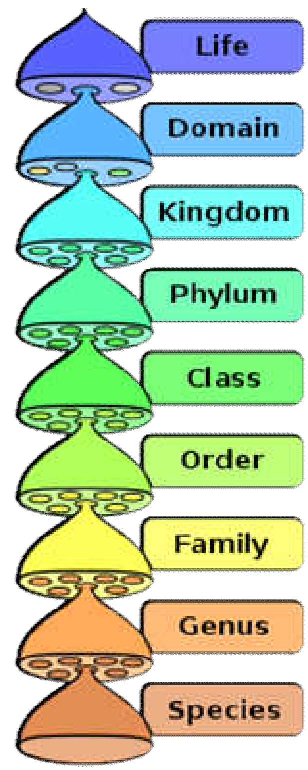
$\text{species, genus, family, order, class} \subset \mathcal{P}(\text{animal})$

$\text{genus} = \mathcal{P}(\text{species})$

$\text{family} = \mathcal{P}(\text{genus})$

$\text{order} = \mathcal{P}(\text{familly})$

$\text{class} = \mathcal{P}(\text{order})$



Examples of some mereotopologic formalisms

- Overlaps, spatial relationships (mereotopology)

Parthood $xPy \equiv x \textit{ is a part of } y$

Proper part x is a proper part of y $x\langle P\rangle y \equiv xPy \wedge \neg yPx$

P and $\langle \mathrm{P} \rangle$ are transitive: $xPy \wedge yPz \Rightarrow xPz$

$aPb \wedge a \neq b \Rightarrow \neg bPa;$

P is antisymmetric: $xPy \wedge yPx \Leftrightarrow x = y$

Overlap proposition $xOy \Leftrightarrow \exists z \ni zPx \ \wedge zPy$

Overlap operator: $x \cap y = z_o \ni z_o Px \ \wedge z_o Py \wedge \forall z_i \neq z_o, z_i Px \wedge z_i Py \Rightarrow z_i PPz_o$

Underlap $xUy \ \equiv \ \exists z \ni xPz \ \wedge yPz$

xOy and xUy are reflexive, symmetric, and intransitive

Overlap Associative $aO(bOc) = (aOb)Oc$

- Behaviors -- Sequences, before-after (4D mereotopology)

Before xBy is transitive: $xBy \ \wedge yBz \Rightarrow xBz$

Proper before is irreflexive $\neg u\langle B\rangle u$

Properbeforeisanti-commutative $a\langle B\rangle b \Rightarrow \neg b\langle B\rangle a$

Properties

- Properties and attributes of classes

Define the powerset of A as the set of all subsets of A:

$$\mathcal{P}(A) = \left\{ \left\{ \ \right\}, \left\{ a_1 \right\}, \left\{ a_2 \right\}, \ldots, \left\{ a_n \right\}, \left\{ a_1, a_2 \right\}, \left\{ a_1, a_3 \right\}, \ldots, \left\{ a_1, a_n \right\}, \ldots, \left\{ a_1, a_2, a_3 \right\}, \ldots \right\}$$

Then:

$B \subset A \Rightarrow B \in \mathcal{P}(A)$

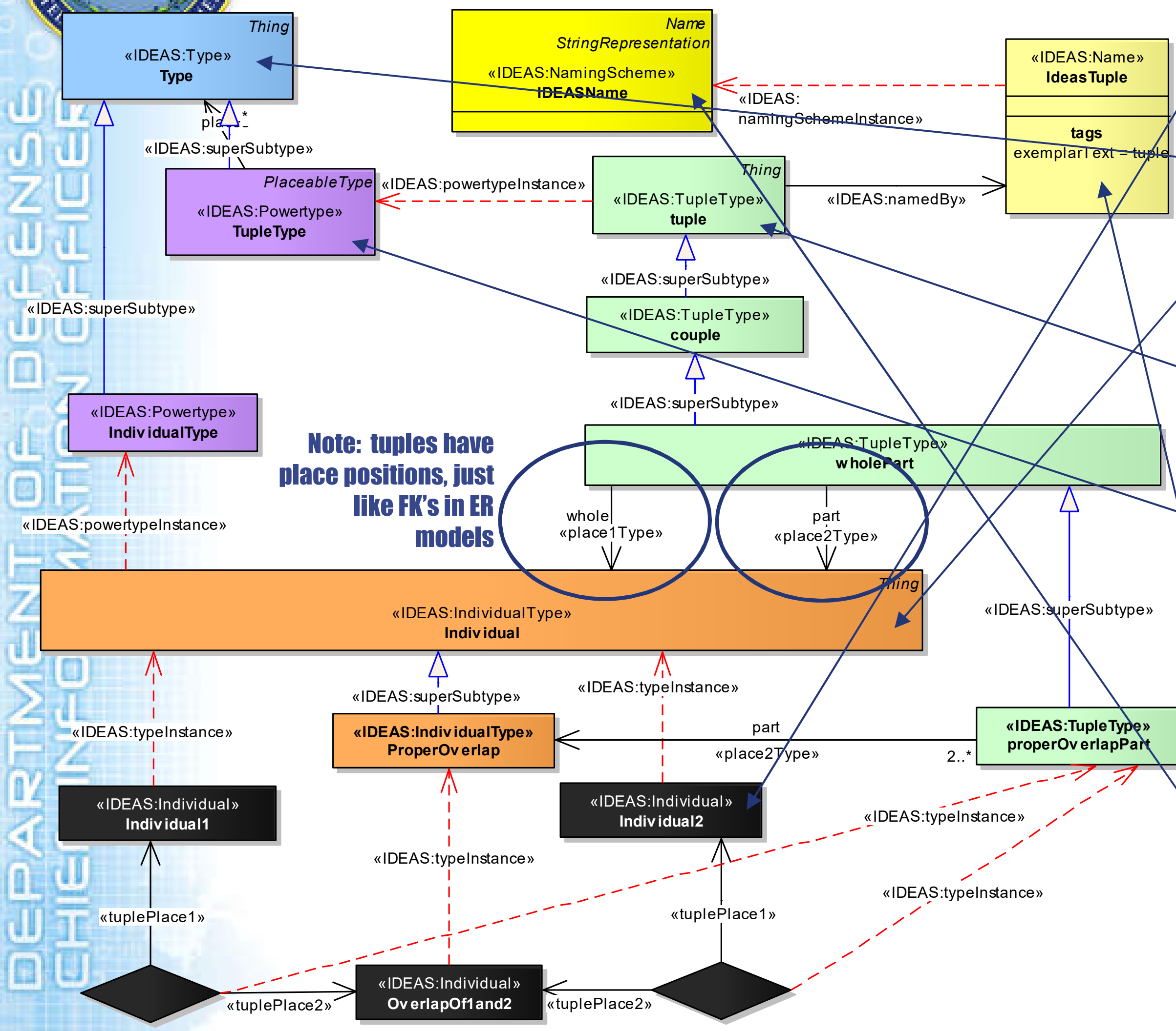
if $\mathbb{A} \subset \mathcal{P}(A) \ni \forall a_m \in A \exists A_i \in \mathbb{A} \ni a_m \in A_i$

then $\mathbb{A}$ is called a "property-of" A or A "has" $\mathbb{A}$

If $\mathbb{A} \equiv \left\{ A_i \right\}, A_i \subset A \ni \mathbb{A}$ is a partition over A

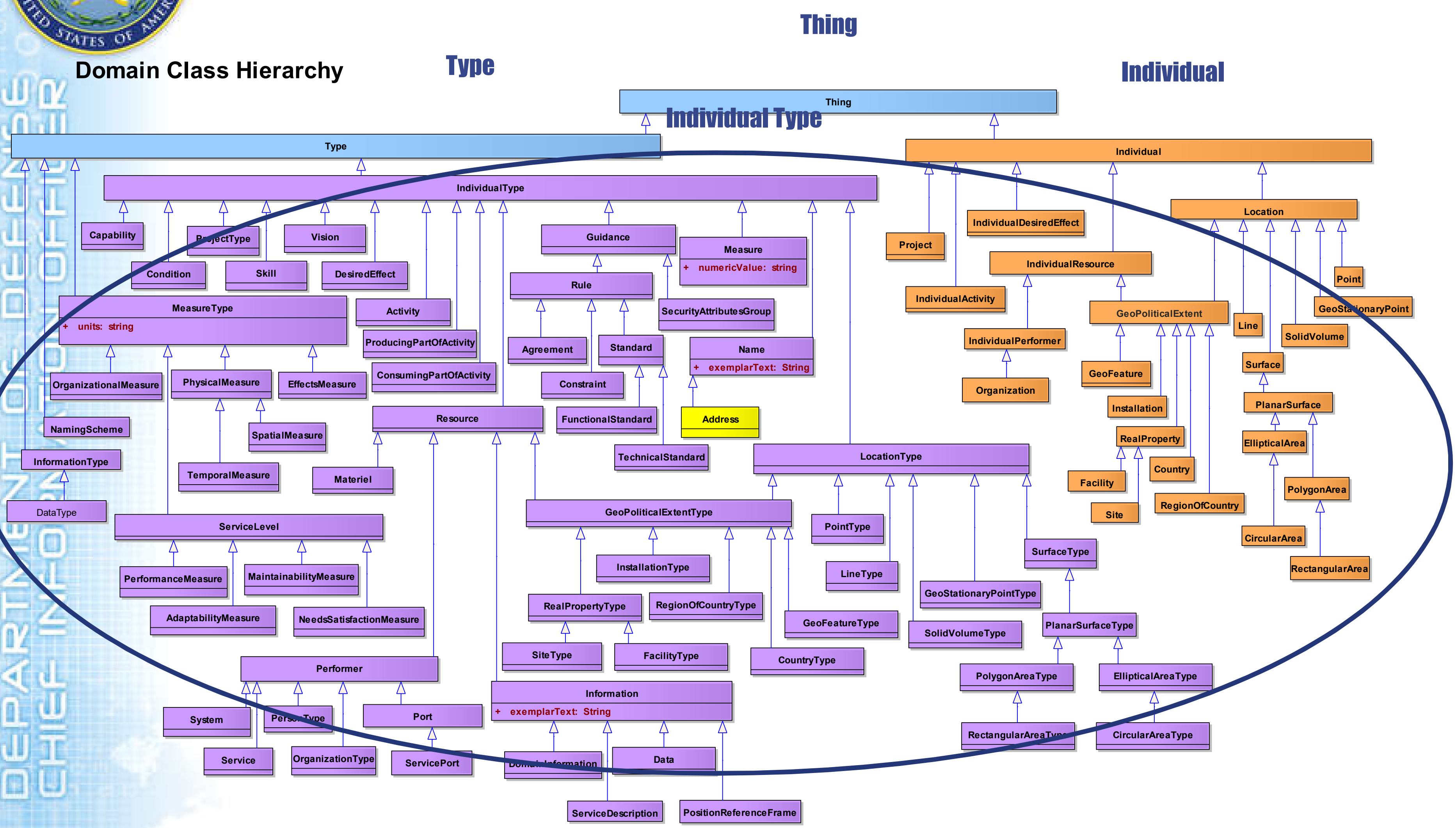
then $\mathbb{A}$ is called a "unique property-of" A

Diagram Conventions and Use of UML



- 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 examplar text provided as a tagged value
- NamingScheme -- The specification of a Type whose members are names

DoDAF Domain Concepts are Specializations

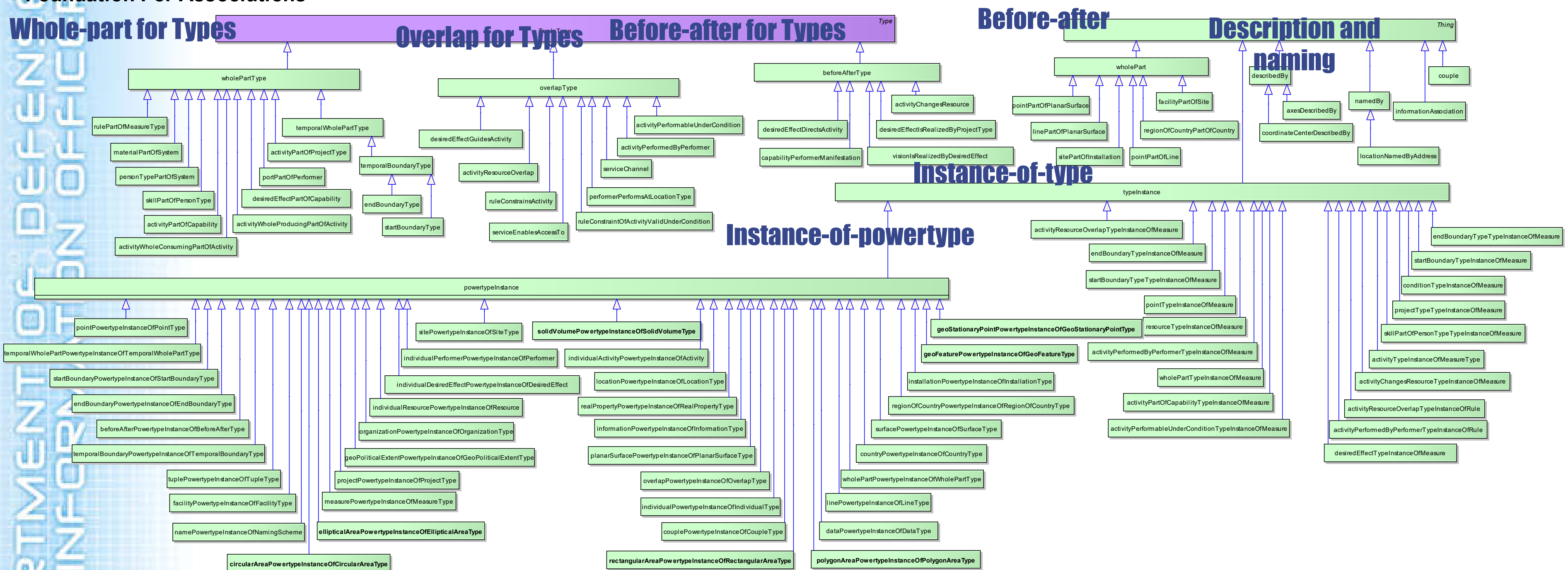


- So they inherit associations (can occupy association place positions)

(zoom-in to read or see handout)

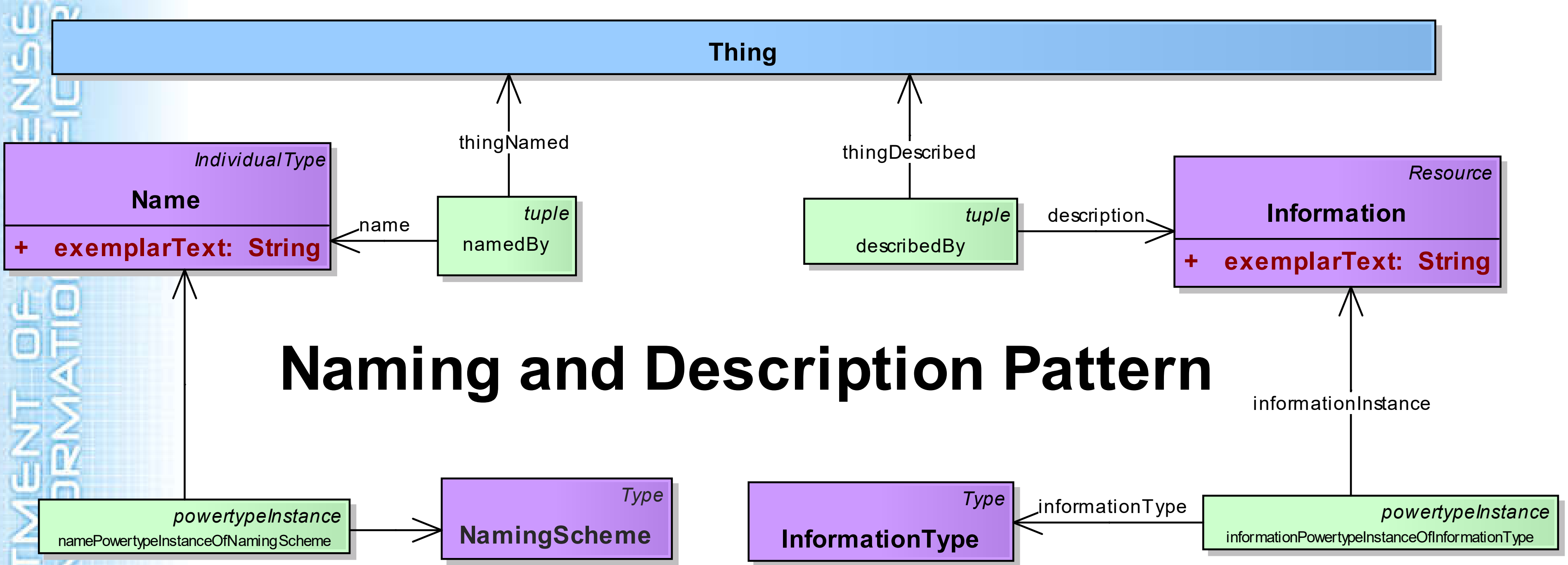
All Associations are Typed

Foundation For Associations



- So their mathematical meaning is formally modeled – a first in DoDAF meta models

Naming and Description Pattern



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

Physical Exchange Specification (PES)

Use Cases Identification / Requirements

Why do I exchange EA data?

1. JCIDS
   - JCD / ICD / CPD / CDD / FNA / FSA / FNA / AoA / TDS Evaluator – overlap and best value comparison
   - ISP / TISP Evaluator – interoperability comparison
   - Tester – “derive” / trace TEMP to
   - Preparer -- reuse
2. DAS
   - Milestone Reviews
   - Gate reviews
   - Functional Control Boards (FCB)
3. PPBE
   - Investment Review Boards (IRB)
   - OMB 300
   - Determine & defend FYDP
4. CPM / CPIC
   - Functional alignment of portfolio
   - PPBE support
5. Systems Engineering
   - Spec development
6. Ops Planning
   - Plans development
   - Interoperability Assurance

Mapping of Models Basis for XSDs

DoDAF models (52)

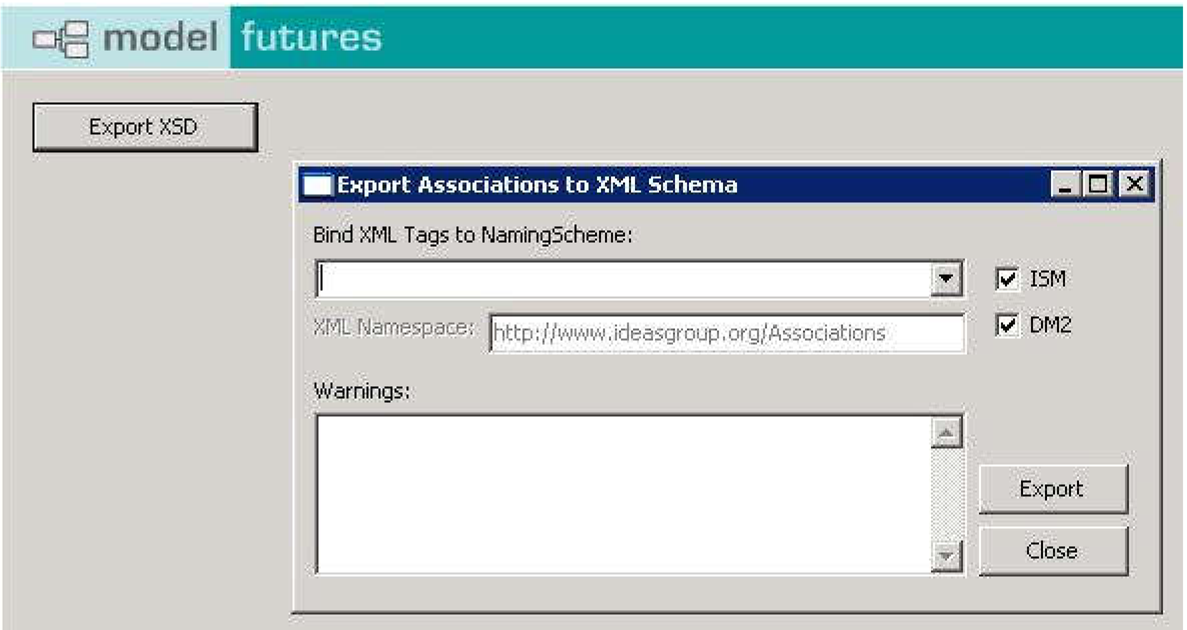
DM2 elements (~ 300)

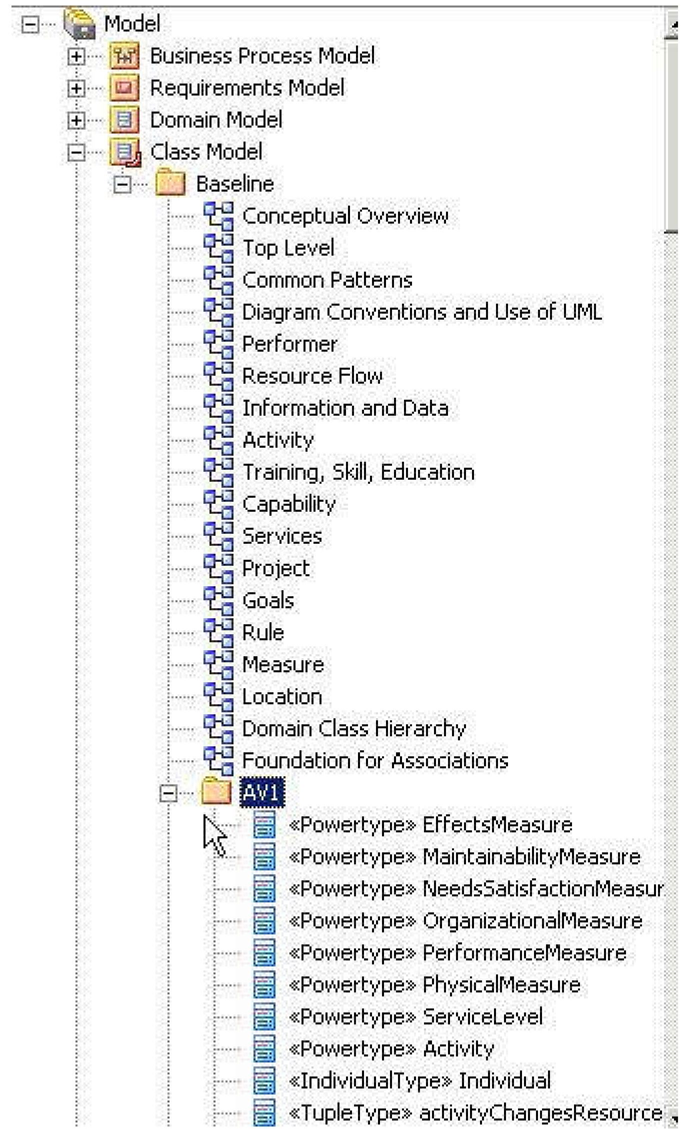
Technical Term	AV-1	AV-2	OV-1	OV-2	OV-3	OV-4	OV-5a	OV-5b	OV-6a	OV-6b	OV-6c	SV-1	SV-2	SV-3	SV-4	SV-5a	SV-5b	SV-6	SV-7	SV-8	SV-9	SV-10a	SV-10b	SV-10c	SvcV-1	SvcV-2	SvcV-3a	SvcV-3b	SvcV-4	SvcV-5	SvcV-6
Activity	n	o		n	n		n	n	o	n	n	n	n	n	n	n	o	n	o	o	o	o	n	n	n	n	n	n	n	o	n
activityC[illegible]	o								o	o	o											o	o	o							
activityChangesResourceTypeInstanceOfMeasure	o								o	o	o											o	o	o							
activityPartOfCapability																	o												o		
activityPartOfCapabilityTypeInstanceOfMeasure																	o														
activityPartOfProjectType																															
activityPerformableUnderCondition								o	o	o	o				o		o					o	o	o							
activityPerformableUnderConditionTypeInstanceOfMeasure								o	o	o	o				o		o					o	o	o							
activityPerformedByPerformer	o			o	o			o	o	o	o				o	o	o	o	o	o	o	o	o	o					o		o
activityPerformedByPerformerTypeInstanceOfMeasure	o		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
activityPerformedByPerformerTypeInstanceOfRule			[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	o
activityResourceOverlap			[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	n
activityResourceOverlapTypeInstanceOfMeasure					o			o	o	o	o	o	o	o	o			o				o	o	o	o	o	o	o			o
activityResourceOverlapTypeInstanceOfRule					o			o	o	o	o	o	o	o	o			o				o	o	o	o	o	o	o			o
activityTypeInstanceOfMeasureType	o				o			o	o	o	o	o	o	o	o			n	n	o	o	o	o	o	o	o					n
activityWholeConsumingPartOfActivity				n	n		o	n	o	n	n	n	n	n	n			n				o	n	n	n	n	n	n			n
activityWholeProducingPartOfActivity				n	n		o	n	o	n	n	n	n	n	n			n				o	n	n	n	n	n	n			n
AdaptabilityMeasure		o			o			o	o	o	o	o	o	o	o			o	o	o	o	o	o	o	o	o	o	o			o
Address	o											o	o					o				o	o		o	o					o
Agreement	o	o						o	o	o	o																				
axesDescribedBy									o			o	o									o			o	o					
beforeAfter	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f
beforeAfterPowertypeInstanceOfBeforeAfterType	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f
beforeAfterType	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f
Capability		o															o														

(see DoDAF Vol II to read entire matrix)

- Legend:
  - "n" = Necessary data for this DoDAF model
  - "o" = Optional
  - "f" = Foundational
  - Blank = cannot be included in this DoDAF model
- **Governance dictates DoDAF models; matrix then dictates what data those models must or can contain**

Exporter


model futures
Export XSD
Export Associations to XML Schema
Bind XML Tags to NamingScheme:
ISM
DM2
XML Namespace:
http://www.ideasgroup.org/Associations
Warnings:
Export
Close


Model
Business Process Model
Requirements Model
Domain Model
Class Model
Baseline
Conceptual Overview
Top Level
Common Patterns
Diagram Conventions and Use of UML
Performer
Resource Flow
Information and Data
Activity
Training, Skill, Education
Capability
Services
Project
Goals
Rule
Measure
Location
Domain Class Hierarchy
Foundation for Associations
AV1
«Powertype» EffectsMeasure
«Powertype» MaintainabilityMeasure
«Powertype» NeedsSatisfactionMeasur
«Powertype» OrganizationalMeasure
«Powertype» PerformanceMeasure
«Powertype» PhysicalMeasure
«Powertype» ServiceLevel
«Powertype» Activity
«IndividualType» Individual
«TupleType» activityChangesResource

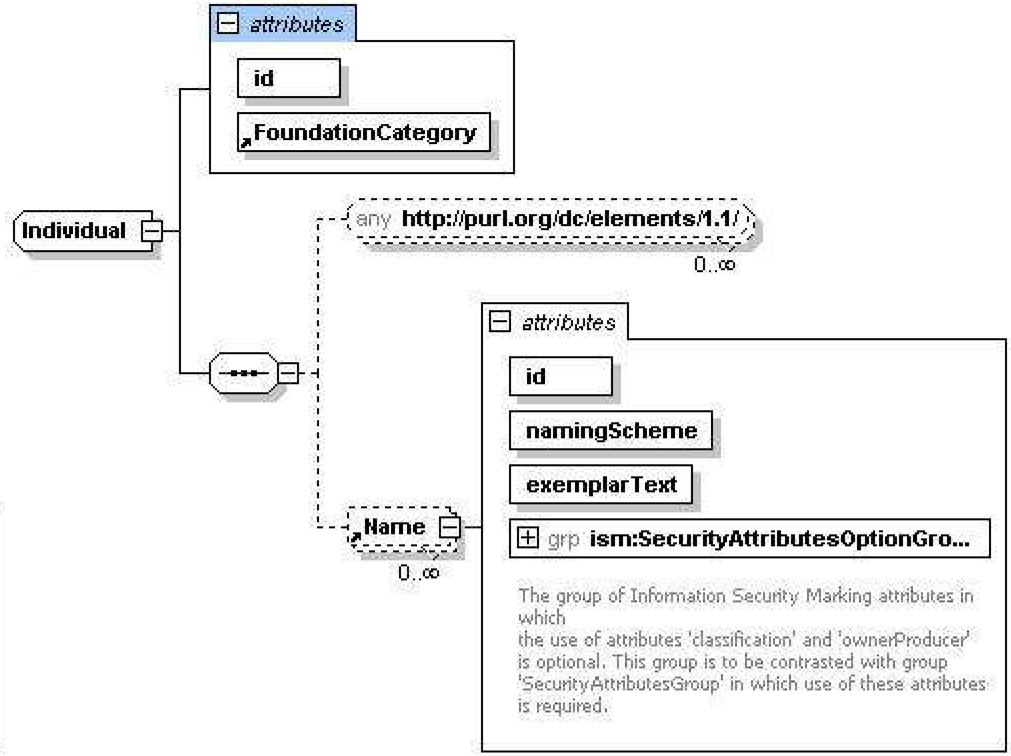

IDEAS Plug-In for Sparx EA
developed by:
model futures
www.modelfutures.com
Development sponsored by UK MOD
(under sub-contract to Serco Consulting)
ideas group
This Plug-In uses the Model Futures Desktop
Ontology Engine (Demonstration License)

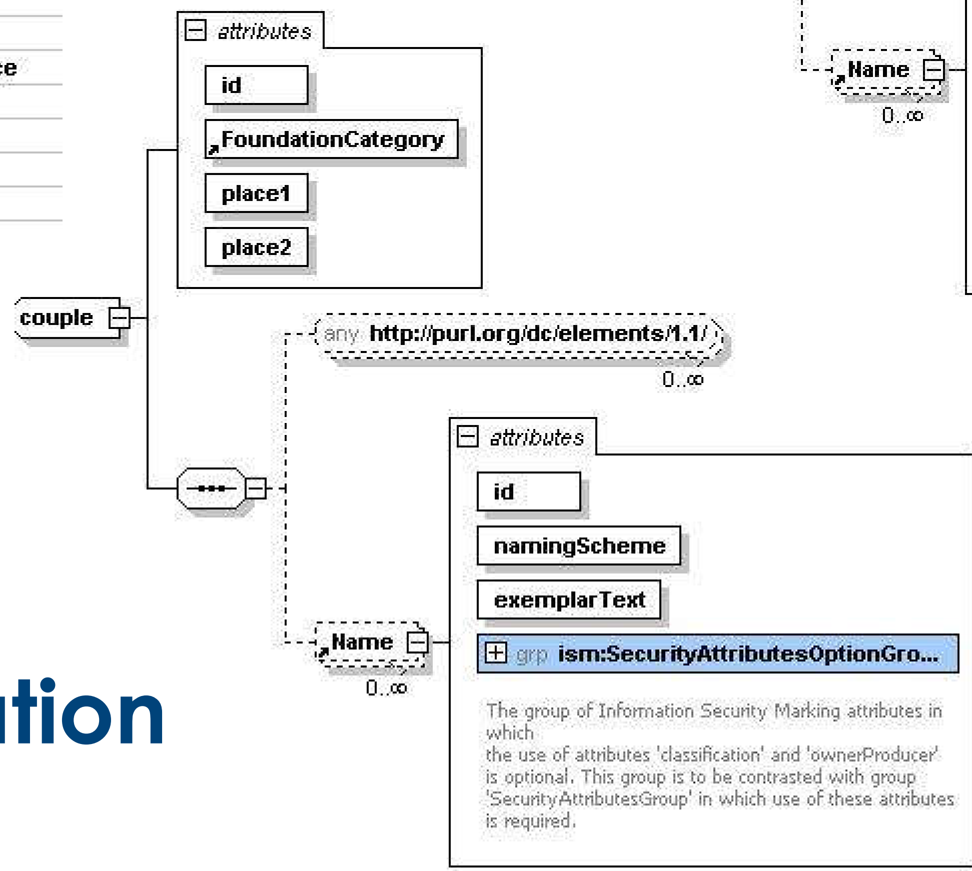

Enterprise Architect
Version 7.1
© 1998 - 2008 Sparx Systems. All rights reserved.
SPARX SYSTEMS

Components

- One per DoDAF model (52) with necessary and optional parts
- 1 comprehensive with all optional for “fit for purpose” models
- 3 references – IDEAS Foundation, Security marking (IC-ISM), and Pedigree
- Physical Exchange Specification (PES) XSD General Structure
  - Wrapper, describing that the document is
  - Independent entities with naming and aliases
  - Associations
  - Constraints
  - Similar to UCORE
- Every piece of data:
  - is tied to the IDEAS Foundation
  - has a classification marking – a “portion mark”
  - has a pedigree (chainable) – who, how,… it came into being

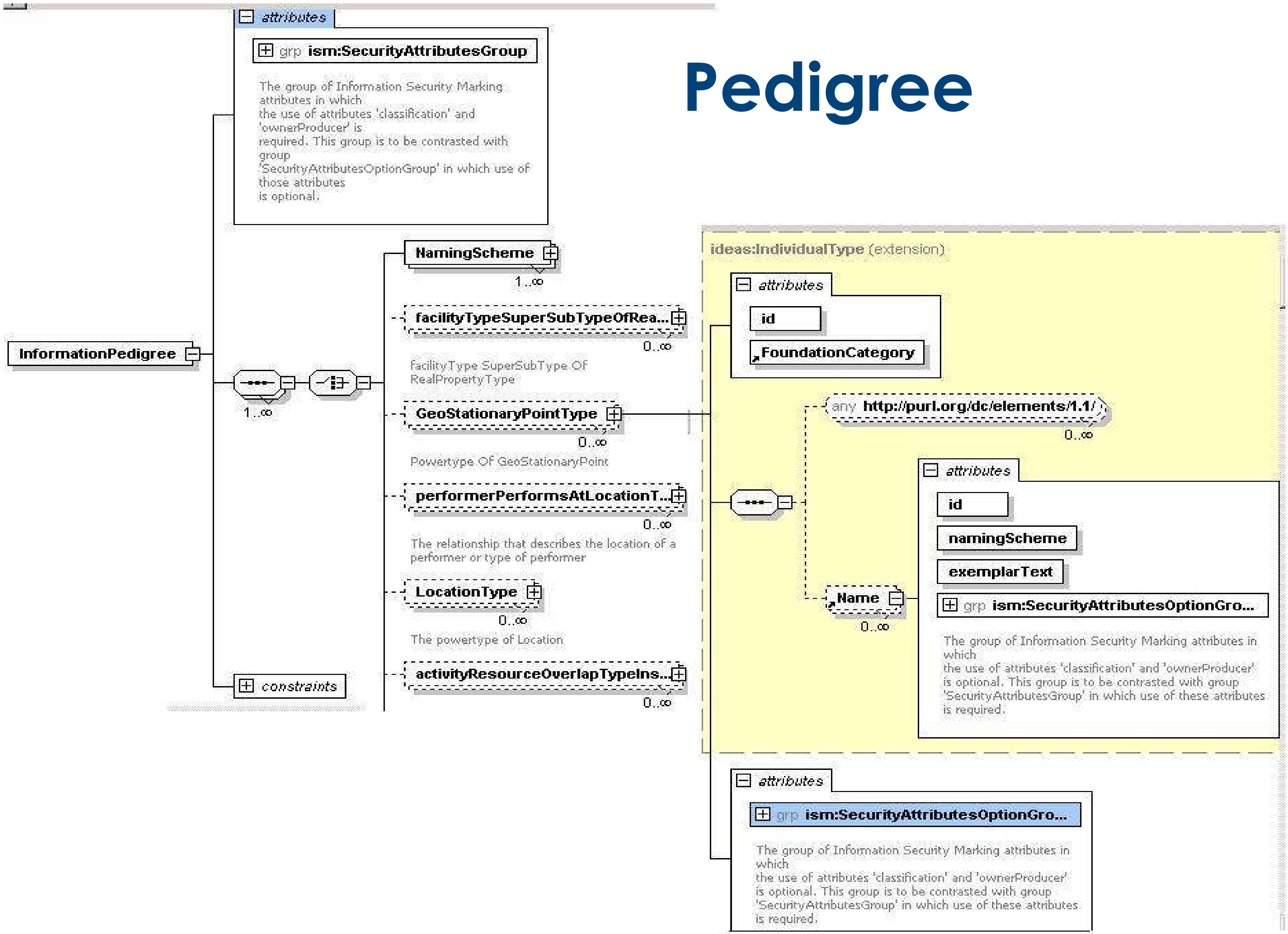
import	loc IC-ISM-v2.1.xsd
simpleType	refList
simpleType	nameList
attribute	FoundationCategory
element	Name
complexType	Thing
complexType	Individual
complexType	Type
complexType	Powertype
complexType	IndividualType
complexType	NamingScheme
complexType	tuple
complexType	couple
complexType	typeInstance
complexType	powertypeInstance
complexType	wholePart
complexType	superSubtype
complexType	triple
complexType	quadruple





IDEAS Foundation

Pedigree


attributes
grp ism:SecurityAttributesGroup
The group of Information Security Marking attributes in which
the use of attributes 'classification' and 'ownerProducer' is
required. This group is to be contrasted with group
'SecurityAttributesOptionGroup' in which use of those attributes
is optional.
InformationPedigree
1..∞
NamingScheme
1..∞
facilityTypeSuperSubTypeOfRea...
0..∞
facilityType SuperSubType Of RealPropertyType
GeoStationaryPointType
0..∞
Powertype Of GeoStationaryPoint
performerPerformsAtLocationT...
0..∞
The relationship that describes the location of a performer or type of performer
LocationType
0..∞
The powertype of Location
activityResourceOverlapTypeIns...
0..∞
constraints
ideas:IndividualType (extension)
attributes
id
FoundationCategory
any http://purl.org/dc/elements/1.1/
0..∞
attributes
id
namingScheme
exemplarText
grp ism:SecurityAttributesOptionGro...
Name
0..∞
The group of Information Security Marking attributes in which
the use of attributes 'classification' and 'ownerProducer'
is optional. This group is to be contrasted with group
'SecurityAttributesGroup' in which use of these attributes
is required.
attributes
grp ism:SecurityAttributesOptionGro...
The group of Information Security Marking attributes in which
the use of attributes 'classification' and 'ownerProducer'
is optional. This group is to be contrasted with group
'SecurityAttributesGroup' in which use of these attributes
is required.

Definition of Terms

- Pedigree: information lineage
  - a chain of sets of observations or object beliefs used to derive the information along with a description of the derivation (i.e., how the set of observations or object beliefs were used)
  - this definition of Pedigree includes the information’s Provenance, that is Provenance Ì Pedigree
- Source Metadata: information about the source
  - a characterization of the source, whether it be a sensor, individual operators, or a system of machines and operators
  - related to Pedigree but is information about the Source, not the particular piece of information being asserted.

P&SM Notes

- P&SM chains can be traversed bi-directionally
  - P&SM lineage describes how a piece of information came about
  - P&SM descendancy describes how a piece of information was used.
- Context could be important in Pedigree
  - “Environmental Context” – what background information did I take into account in deriving this information? (And how did I derive that information, i.e., what was my Context estimation P&SM.)
  - “Mission Context” – what am I doing that focusses and influences the way I perceive things
- There can be different levels of detail or granularity of P&SM for different purposes.
  - In DM2, we provided for the most granular, allowing user to use aggregates as necessary
- IA / Security
  - Source Metadata must allow for information sharing while protecting sources and methods

Ascendant Use Notional Examples

Aid Information Management -- being able to pull the lineage and source information aids understanding of the information so fusion nodes can better collaborate in developing and sharing knowledge to achieve situation awareness, e.g.,

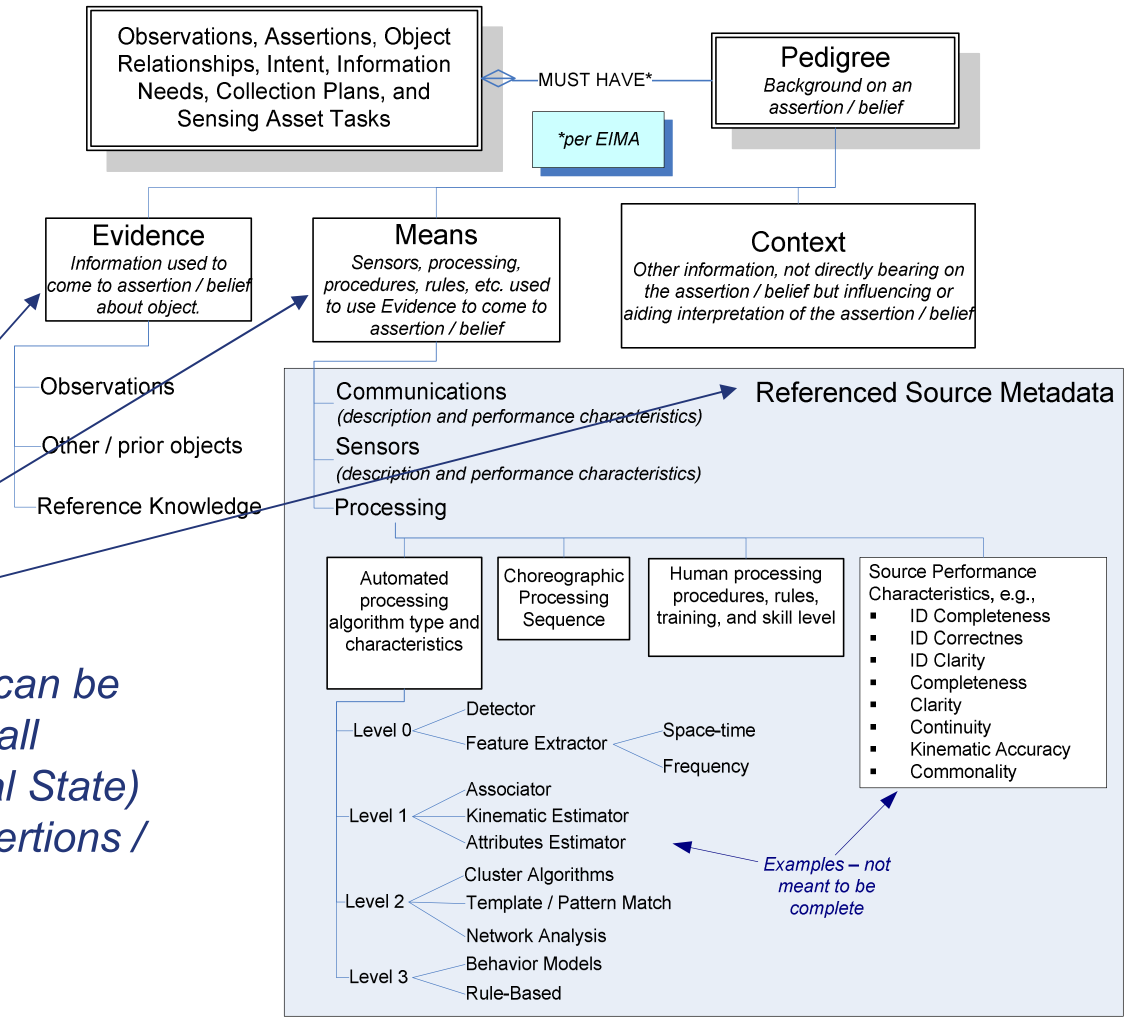
- Integration of multiple sources may require P&SM for reconciliation of differences in assertions / beliefs between fusion nodes. For example, ID Conflict or correlation difference in the TADIL's. These are currently worked off manually via voice circuit. The inefficiencies and dis-interoperabilities inherent in this design have been shown in many Joint exercises.
- Aid in fusing the provided information with other information – P&SM augments the quality of provided information. Although certainty estimates (e.g., confidences, covariances) may be published or available for information, they may not tell the whole story when estimation model assumptions are violated. Interpretation of an assertion or belief.
- Support corroboration analysis[1] and avoid of information double counting[2] (also known as "data incest", "rumor propagation", or "data ringing")
- Collection / Sensor Resource Management. P&SM shows what sources have already be employed and that, therefore, may not have much additional value in re-tasking, thereby avoiding wasted resource utilization.
- Assess the trustworthiness or quality of provided information by pulling the lineage chain and assessing the trustworthiness of the sources. (How does "trust" apply to systems / machines and how does it differ from reliability, accuracy, ….?)
- Removal of aberrancies. The P&SM chain may have to be pulled to re-estimate the object.

- Aid Force Protection and Engagement
  - Being able to pull P&SM information on a object being targeted could provide quicker confidence that a target should be engaged prior to the engagement, somewhat analogous to the Mode 4 pre-engagement interrogation.
  - After the engagement, it may be useful to pull P&SM information in conducting kill assessment using multiple sources of information of disparate types.
- Aid Operational Planning
  - It may be valuable in formulating and evaluating alternative maneuver Courses of Action, being able to reach back into critical pieces of Situation Awareness data whose accuracy and interpretation could "swing" a CoA decision. Knowing the trustworthiness and accuracy of information could
    - alert the maneuver commander to monitor conditions of interest once the maneuver is underway
    - help pre-formulate risk mitigation alternative CoAs.

Descendant Use Notional Examples

- Understand how published information is used by others
- Remove aberrancies (own-force "mistakes") or deceptions (opposing force) – knowing who got "contaminated"
  - Doesn't tell you indirect effects of the wrong information
- Assess security vulnerabilities from inadvertently disclosed information, e.g., by maintaining traceability for understanding who touched what data in support of vulnerability analysis.

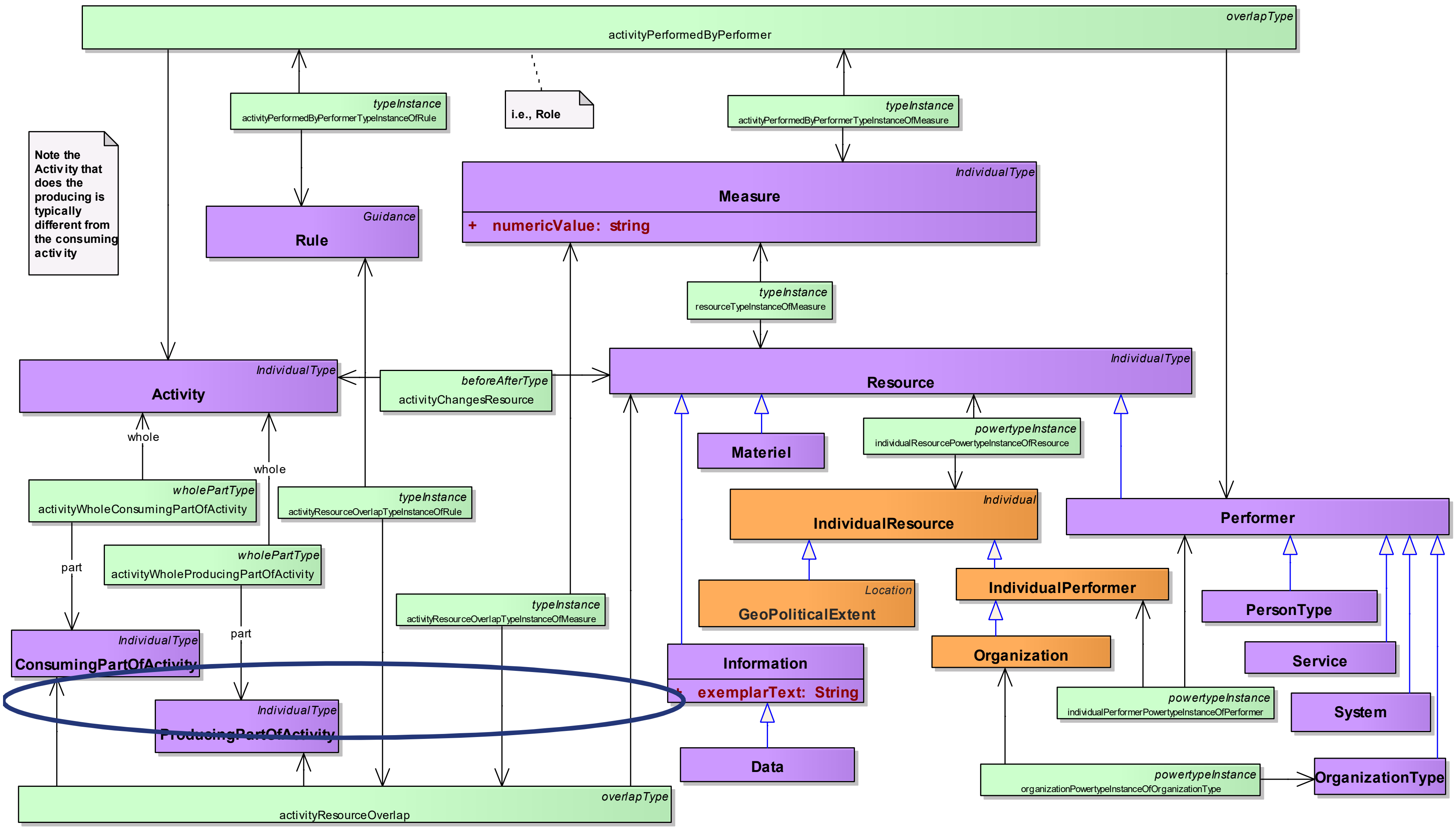
Components of P&SM



Important point: P&SM can be associated with the overall Thing (or Thing Temporal State) as well as individual assertions / beliefs about the Thing

Pedigree Model Based on DM2 Resource Flow Model

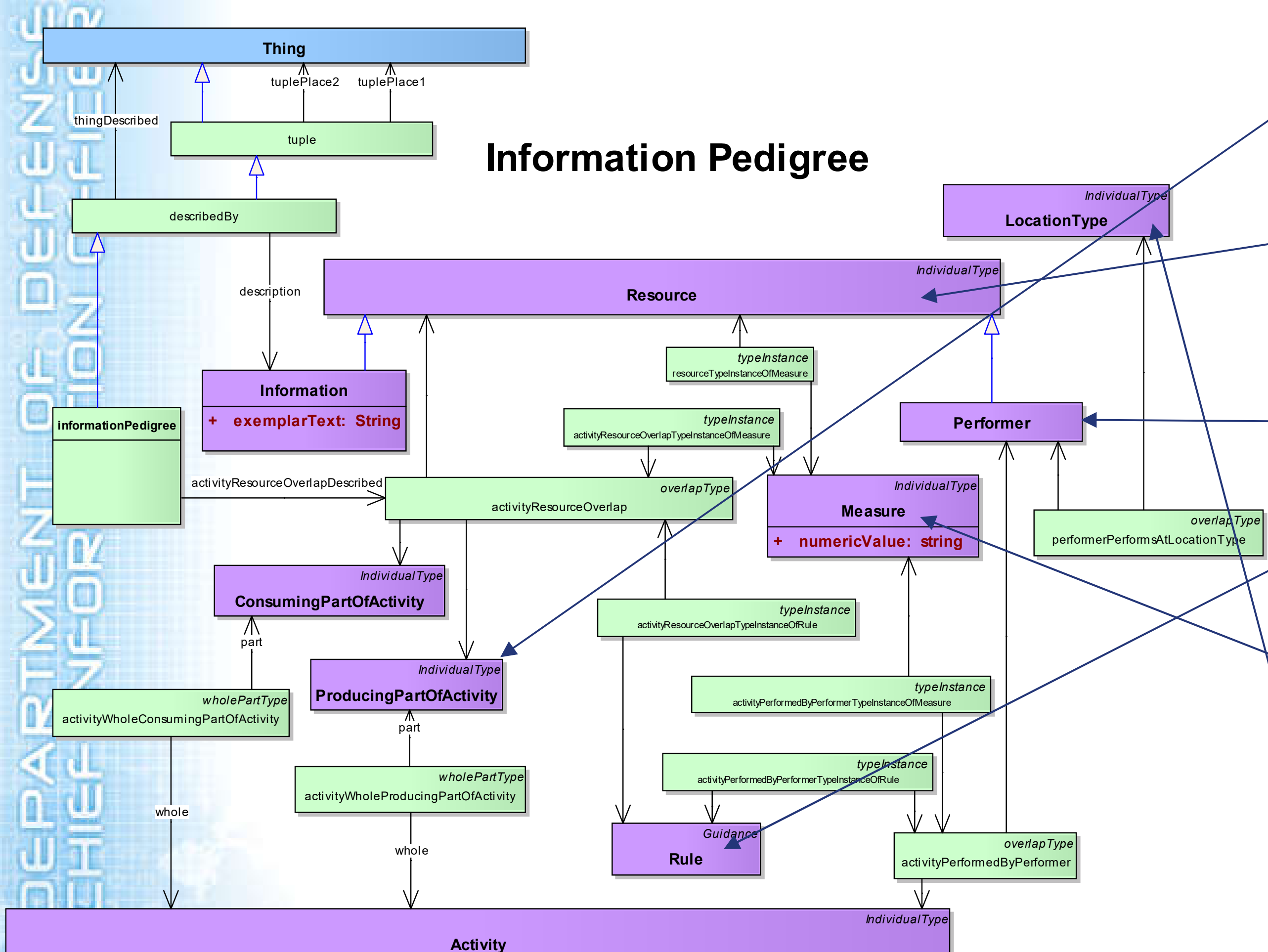
Resource Flow



Resource Flow Model 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*).

Information Pedigree – workflow model
~ Open Provenance Model (provenance = linked together pedigrees)



Information Production Activity

Resources Used, e.g., other Information

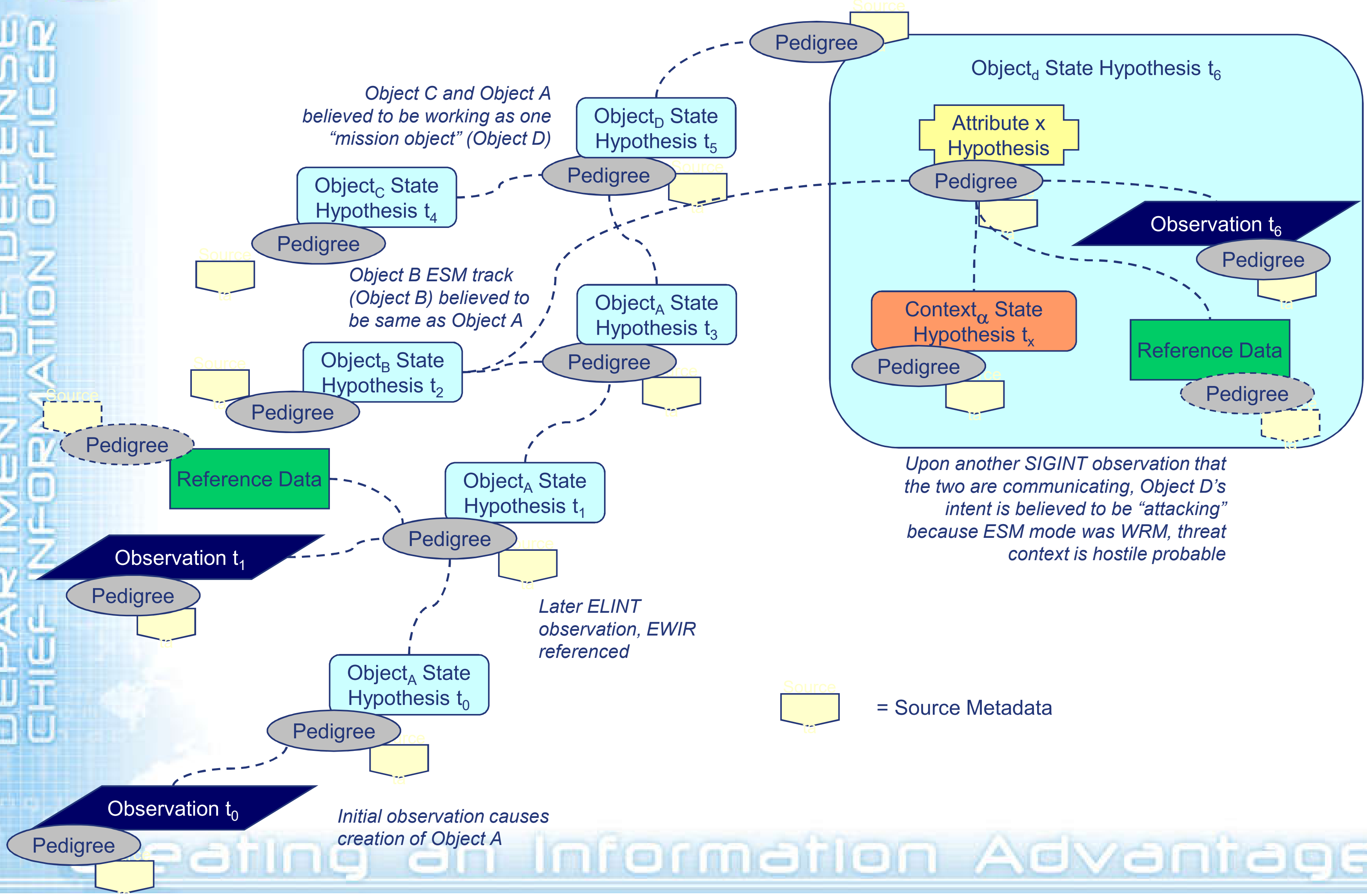
Who

Rules followed in the production

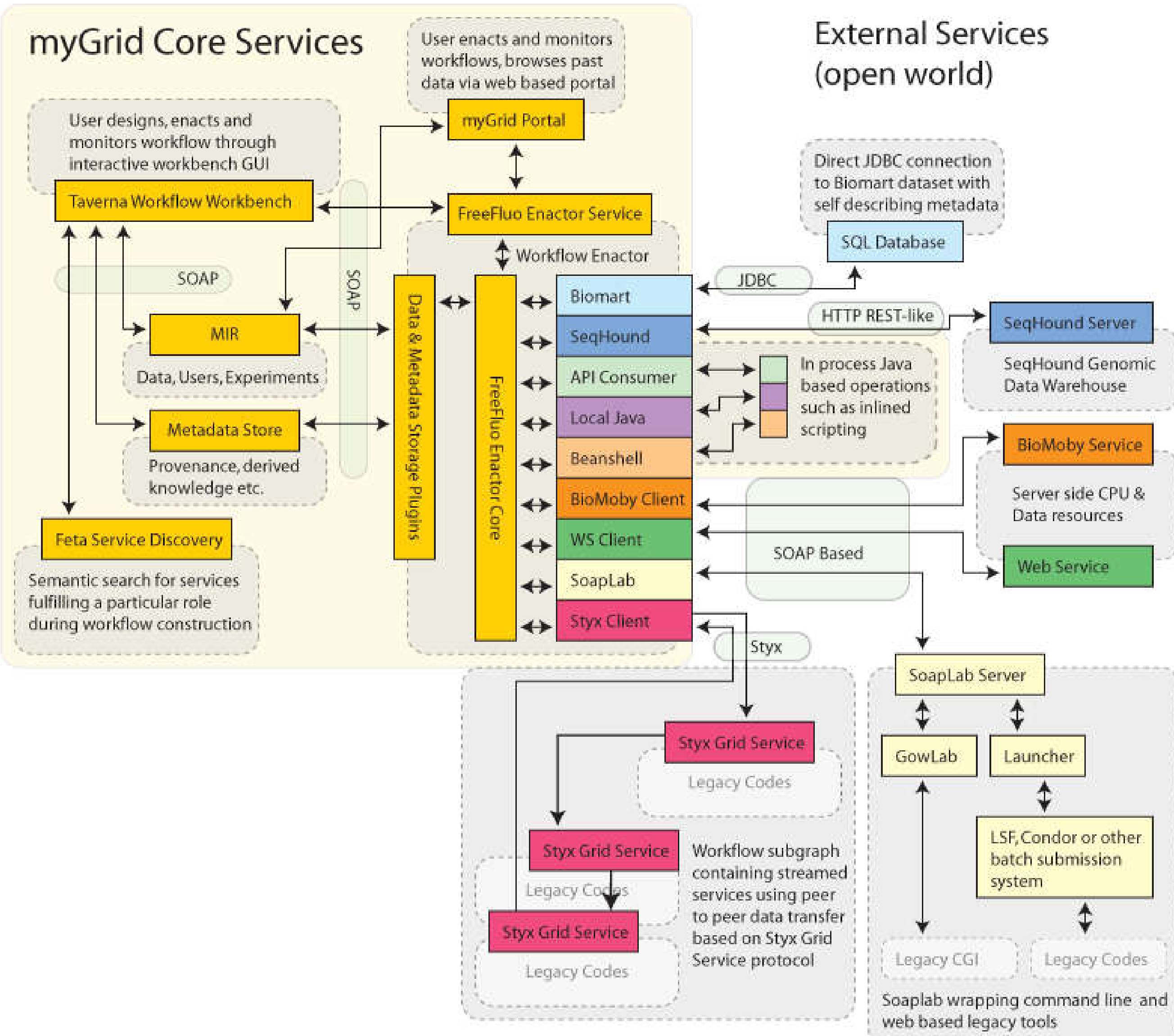
Measures in the production, e.g., QoS, uncertainties

Where done

Pedigree Chaining



Example of Pedigree and Workflow in Biologic Research Community



- Taverna network architecture diagram

DEPARTMENT OF DEFENSE
CHIEF INFORMATION OFFICER

Questions?