model futures
www.modelfutures.com

presents:

IntegratedEA

STRATEGY • OPERATIONS • TECHNOLOGY

www: http://www.integrated-ea.com
HashTag: #IEA14
Twitter: @IntegratedEA

Ministry of Defence
CESG
mega
GCHQ
de&S
IBM
BMT Hi-Q Sigma
"Where will our knowledge take you?"
MooD
SMARTER DECISIONS
Reply
glue
JCSys
ATKINS
SCJ SecurityCleared Jobs.com
UKCeB

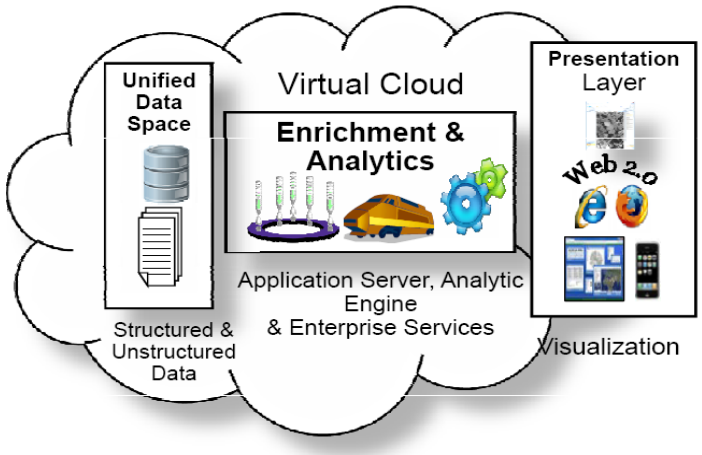
Warfighter Mission Area

Information Mission Area

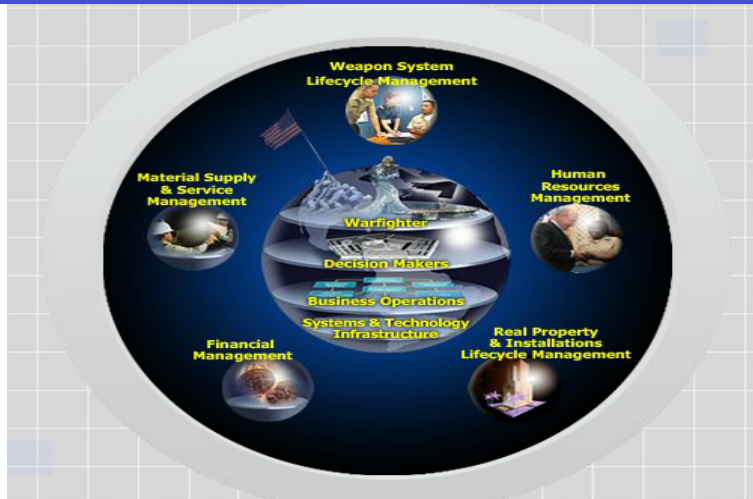
Lines of Sight and Provable Traceability

Presentation by Mr. David McDaniel and Gregory Schaefer
Silver Bullet Solutions, Inc.

2014 Integrated EA Conference



Intelligence Mission Area



Business Mission Area

Presentation Outline

- Why traceability is important
- Issues with traceability
- Ontology and predicate calculus of traceability
- Application to architectural patterns
- Summary and possible future work

Why traceability is important

Cartoon

Traceability is flip side of reification

JIE example

The old, “what the customer wanted”

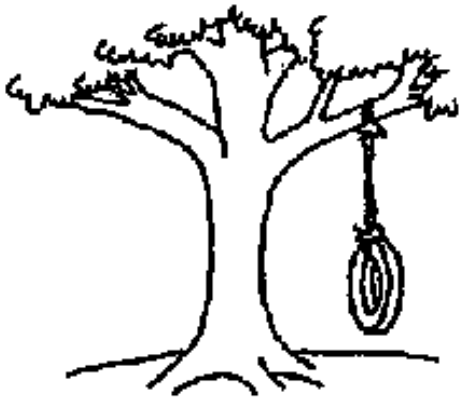
As proposed by the project sponsor.

As specified in the project request.

As designed by the senior analyst.

As produced by the programmers.

As installed at the user's site.



What the user wanted.

Traceability = Reification^{-1}

AN ARCHITECTURAL DESCRIPTION:

IDENTIFIES

Traceability (Pedigree)

Rules constrain (requirements)

DEFINES

Traceability (Pedigree)

Rules constrain (requirements)

REPRESENTS

Traceability (Pedigree)

Rules constrain (requirements)

SPECIFIES

Traceability (Pedigree)

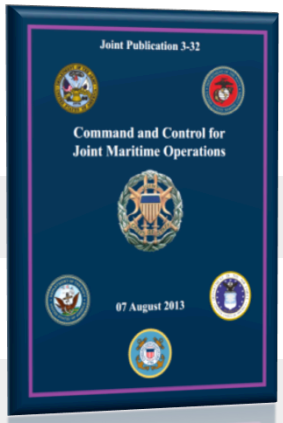
Rules constrain (requirements)

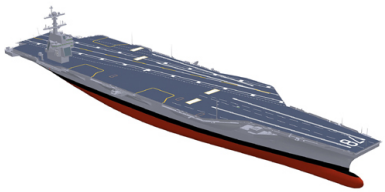
CONFIGURES

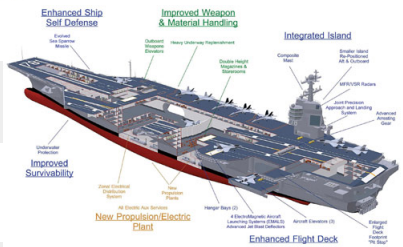
Traceability (Pedigree)

Rules constrain (requirements)

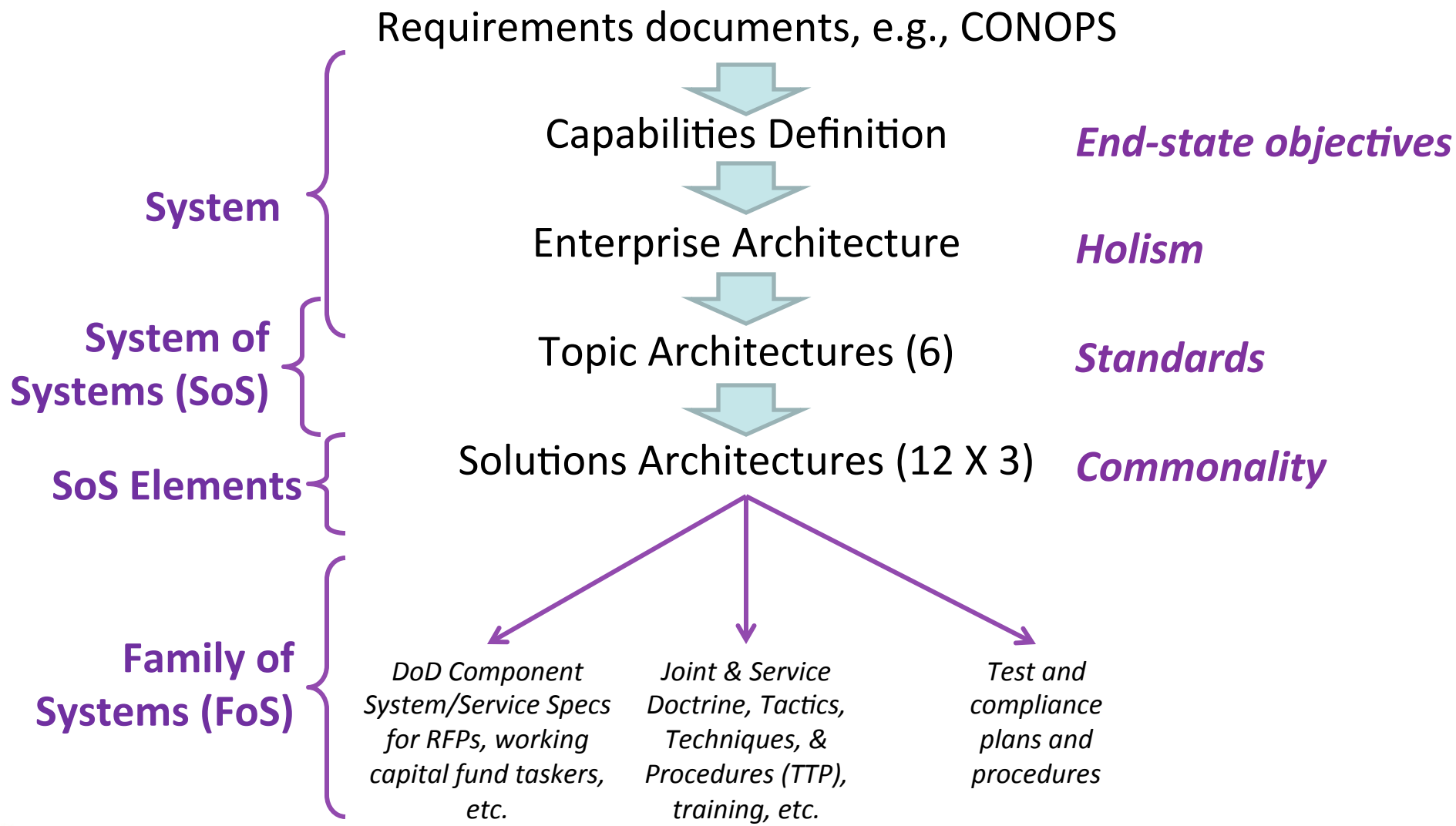
INSTANTIATED







DoD's Joint Information Environment (JIE) Project



Challenges with traceability

Document orientation

Element orientation

Document Oriented

- Defense Acquisition University

List of Sources (include date & revision number)

Requirements Source:
Requirements Source:
Specification Reference:
Design Reference:

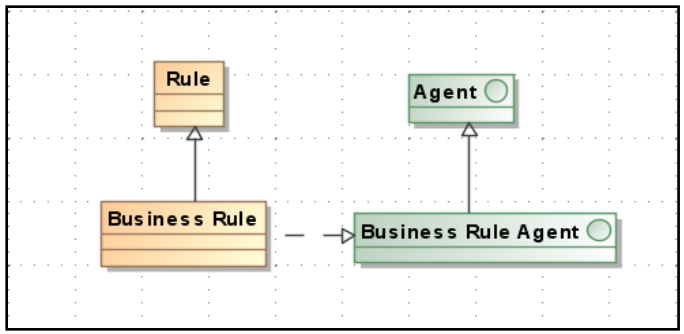
Requirements Extractor:
Requirements Mapper:
Requirements Verifier (Test):
Matrix Reviewer:

Req Id	Requirement Name	Requirement detail	Priority	Test Scenario Name	Req Type	Test Method	Status	Owner	Comments

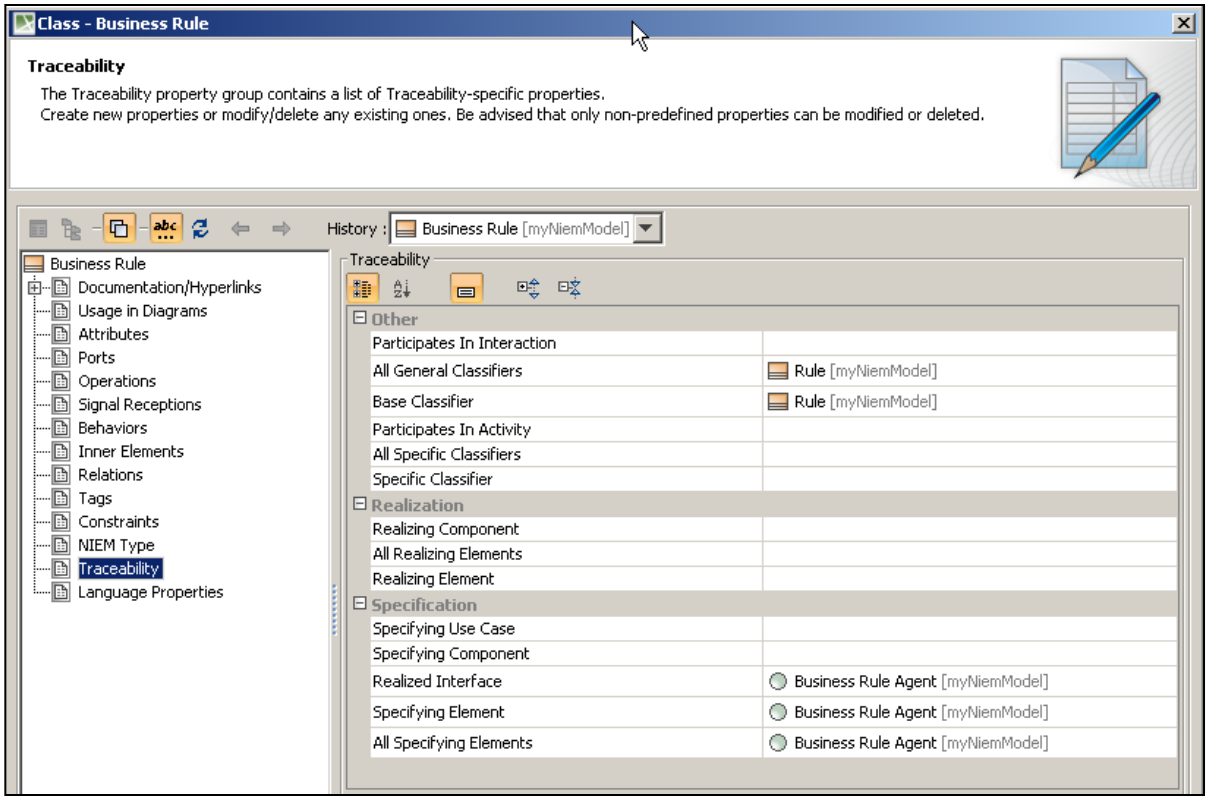
- INCOSE

ID	Assoc ID	Technical Assumption(s) and/or Customer Need(s)	Functional Requirement	Status	Architectural/Design Document	Technical Specification	System Component(s)	Software Module(s)	Test Case Number	Tested In	Implemented In	Verification	Additional Comments
001	111												
002	2.2.2												
003	3.3.3												
004	4.4.4												

MBSE Tools



- Many relationships across many diagrams
- Traceability is a embedded menu per element



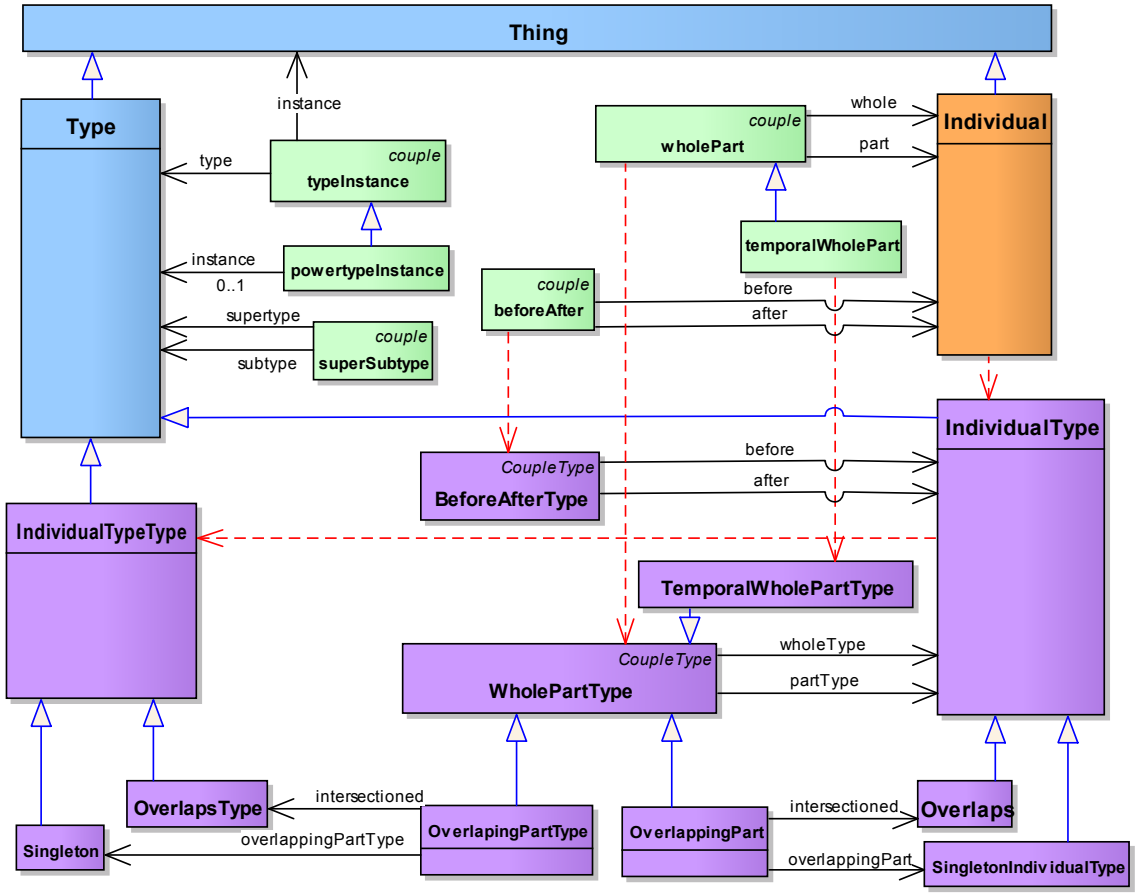
Ontology and predicate calculus of traceability

International Defence Enterprise Architecture Specification (IDEAS)

Predicate calculus of key IDEAS relationships

IDEAS Summary

- International Defence organizations
- Upper level consistent with SUMO and ISO 15926
- Set theory
- 4-D mereotopology



WholePart: $X \underset{WP}{\subseteq} Y$ and $\mathbf{X} \underset{\mathbf{WPT}}{\subseteq} \mathbf{Y}$

Overlap: $X \underset{OV}{\mathrm{I}} Y$ and $\mathbf{X} \underset{\mathbf{OVT}}{\mathbf{I}} \mathbf{Y}$

BeforeAfter: $X \underset{BA}{\leq} Y$ and $\mathbf{X} \underset{\mathbf{BAT}}{\leq} \mathbf{Y}$

Spatio - temporal merge : $\overset{WP}{\underset{\eta=1,..,n}{\mathrm{U}}} X_{\eta}$ and $\overset{\mathbf{WPT}}{\underset{\mu=1,..,m}{\mathbf{U}}} \mathbf{X}_{\mu}$

The DoDAF Meta Model simply extends from IDEAS

Notation for WholePart

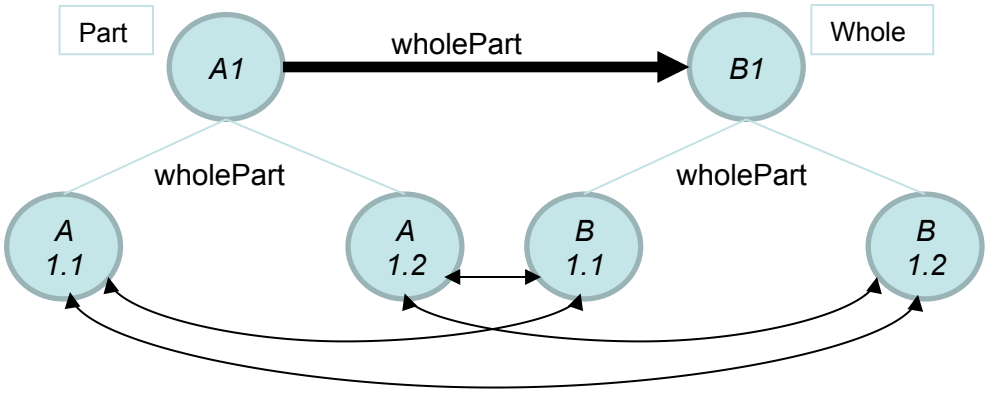
A spatio-temporal extent X is a part of spatio-extent Y, denoted $X \underset{WP}{\subseteq} Y$ iff ∀ point x within X, x is also within Y.

1. A proper part is denoted $X \underset{WP}{\subset} Y$ meaning ∃ y within Y, y not within X
2. $\underset{WP}{\subseteq}$ and $\underset{WP}{\subset}$ are transitive: $X \underset{WP}{\subseteq} Y \ \wedge Y \underset{WP}{\subseteq} Z \ \Rightarrow X \underset{WP}{\subseteq} Z$
3. $\underset{WP}{\subset}$ is antisymetric so $X \underset{WP}{\subseteq} Y \wedge Y \underset{WP}{\subseteq} X \Leftrightarrow \ X=Y$

For two sets of spatio-temporal extents **X** and **Y**, $\mathbf{X}=\{X_i\}, \mathbf{Y}=\{Y_j\}$ whole-part at the type level is

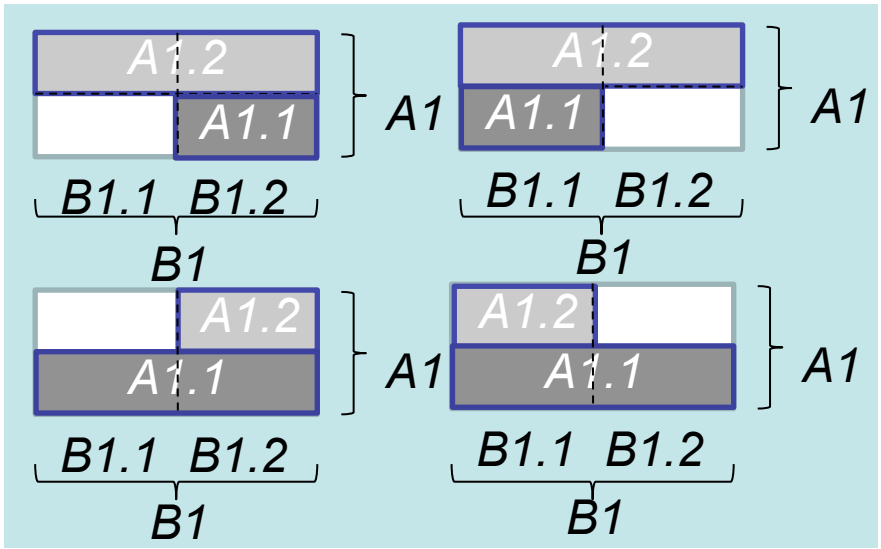
denoted $\mathbf{X} \underset{\mathbf{WPT}}{\subseteq} \mathbf{Y}$ iff $\forall X_i \in \mathbf{X}\ \exists\ Y_j \in \mathbf{Y}\ \ni\ X_i \underset{WP}{\subseteq} Y_j$ and $\forall Y_j \in \mathbf{Y}\ \exists\ X_i \in \mathbf{X}\ \ni\ X_i \underset{WP}{\subseteq} Y_j$

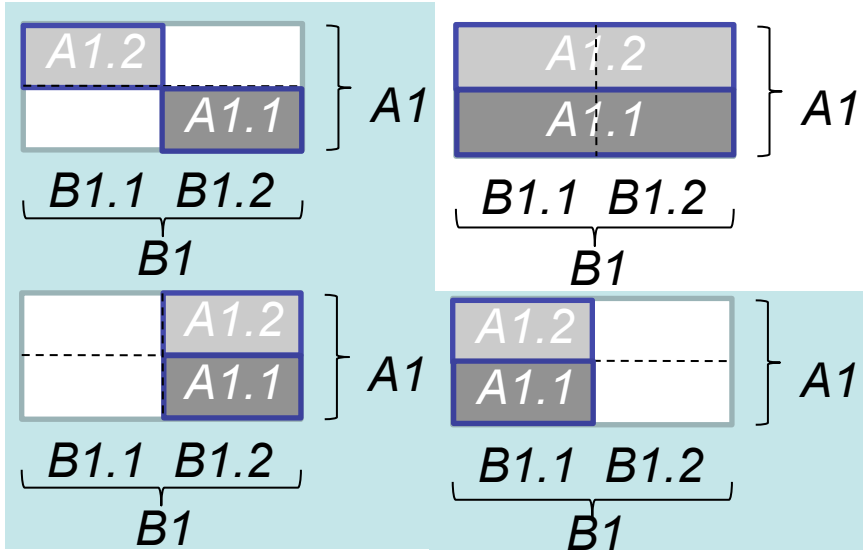
Validation for Relationship Type: wholePart



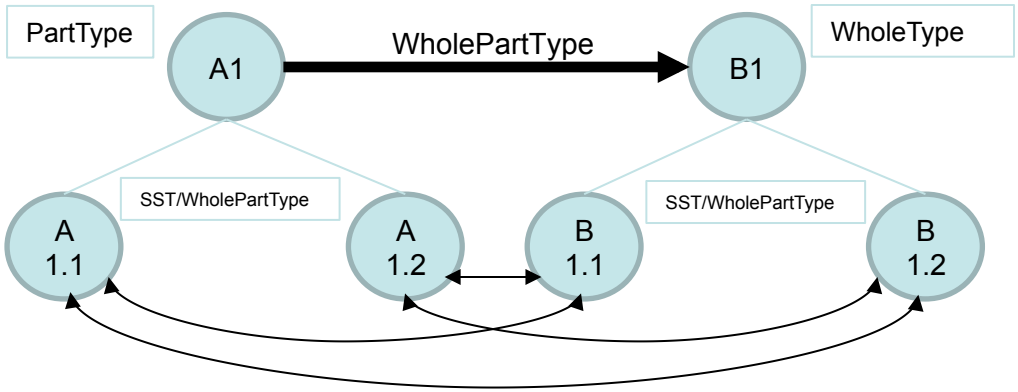
Validation Check:

1. Every part of *A1* must have at least one overlap with a part of *B1*
2. If all the parts of *A1* have a wholePart with the any part of *B1*, then *A1* must be a part of *B1*
3. If any part of *A1* has an overlap with any part of *B1*, then *A1* must overlap *B1*





Validation for Relationship Type: WholePartType



Validation Check:

1. Every SST/WPT of A1 must have a PartType with B1 and conversely, every SST/WPT of B1 has a inverse PartType with A1 (Note 1)
2. If all the PartTypes of A1 have a WholePartType with any PartType of B1, then A1 must be a WholePartType of B1 (and A1 = union A11 and A12 and B1 = B11 and B12)
3. If all SST of A1 have an OverlapType with any SST of B1 and conversely, then A1 must OverlapType B1 (Note 1)
4. If all the PartType of A1 has a WholePartType with any PartType of B1 and conversely, then A1 must OverlapType B1 (Note 2)
5. If all PartTypes of A1 have an OverlapType with any PartType of B1 and conversely, then A1 must OverlapType B1 (Note 2)

Note 1: SST assumes complete subsets
Note 2: WPT assumes complete parts

Notation for Overlap

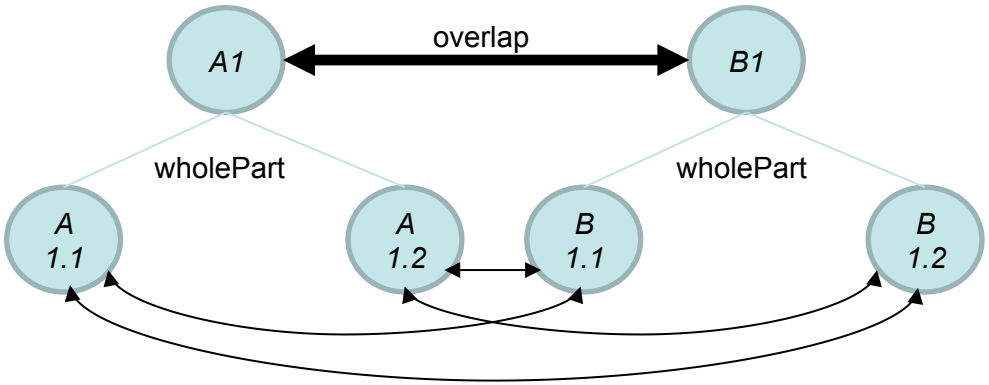
For two spatio-temporal extents X and Y, their Overlap is denoted $X \underset{OV}{\mathrm{I}} Y$. (Note 1)

1. $\left(X \underset{OV}{\mathrm{I}} Y\right) \underset{\mathrm{WP}}{\subseteq} X \wedge \left(X \underset{OV}{\mathrm{I}} Y\right) \underset{\mathrm{WP}}{\subseteq} Y \ \wedge \ \forall Z \ \ni \ Z \underset{\mathrm{WP}}{\subseteq} X \ \wedge Z \underset{\mathrm{WP}}{\subseteq} Y, \ Z \underset{\mathrm{WP}}{\subseteq} \left(X \underset{OV}{\mathrm{I}} Y\right)$
2. $\underset{OV}{\mathrm{I}}$ is reflexive, symmetric, and intransitive
3. $\underset{OV}{\mathrm{I}}$ is associative $X \underset{OV}{\mathrm{I}} (Y \underset{OV}{\mathrm{I}} Z) = (X \underset{OV}{\mathrm{I}} Y) \underset{OV}{\mathrm{I}} Z$

Two sets of spatio-temporal extents X and Y, $\mathbf{X}=\{X_i\}, \mathbf{Y}=\{Y_j\}$ are said to Overlap at the type level, denoted $\mathbf{X} \underset{\mathbf{OVT}}{\mathbf{I}} \mathbf{Y}$ iff $\forall X_i \in \mathbf{X} \ \exists \ Y_j \in \mathbf{Y} \ \ni \ X_i \underset{OV}{\mathrm{I}} Y_j$ and vice-versa.

Note 1: If $X \underset{OV}{\mathrm{I}} Y \neq \varnothing$, X and Y are said to Overlap. Although the same symbol is used to denote both the spatio-temporal extent of the overlap and the binary test of whether there is an overlap, the context will make clear which meaning intended by the $\underset{OV}{\mathrm{I}}$ symbol.

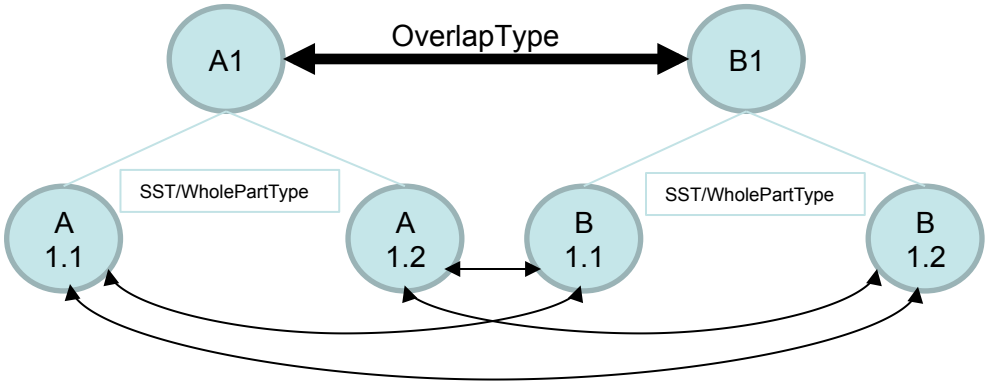
Validation for Relationship Type: overlap



Validation Check:

1. At least one part of *A1* must have at least one overlap with a part of *B1* and at least one part of *B1* must have at least one overlap with a part of *A1*.
2. If any part of *A1* has an overlap with any part of *B1*, then *A1* must overlap *B1* and vice versa.

Validation for Relationship Type: OverlapType



Validation Check:

1. Every SST of A1 must have a OverlapType with B1 and conversely, every SST of B1 has a OverlapType with A1 (Note 1)

2. There must exist a WPT of A1 that has an OverlapType with B1 and conversely, there must exist a WPT of B1 that has a OverlapType with A1

3. If all SST of A1 have an OverlapType with any SST of B1 and conversely, then A1 must OverlapType B1 (Note 1)

Note 1: SST assumes complete subsets

Notation for spatio-temporal merges

For n spatio-temporal extents X_i, their spatio-temporal merge is denoted $\overset{WP}{\underset{\eta=1,..,n}{\mathrm{U}}} X_\eta$

1. $\forall i = 1,..,n,\ X_i \underset{WP}{\subseteq} \left(\overset{WP}{\underset{\eta=1,..,n}{\mathrm{U}}} X_\eta \right)$

For m sets of spatio-temporal extents $\mathbf{X_m}$, $\mathbf{X_m} = \{X_{m,\eta}\}$, merge at the type level, denoted $\overset{WPT}{\underset{\mu=1,..,m}{\cup}} \mathbf{X}_\mu$ means $\forall X_i \in \overset{WPT}{\underset{\mu=1,..,m}{\cup}} \mathbf{X}_\mu \ \exists X_{m,\eta} \in \mathbf{X_m} \ni X_i = \overset{WP}{\underset{\eta=1,..,n}{\mathrm{U}}} X_\eta$ and vice-versa.

Application to architectural patterns

Capabilities

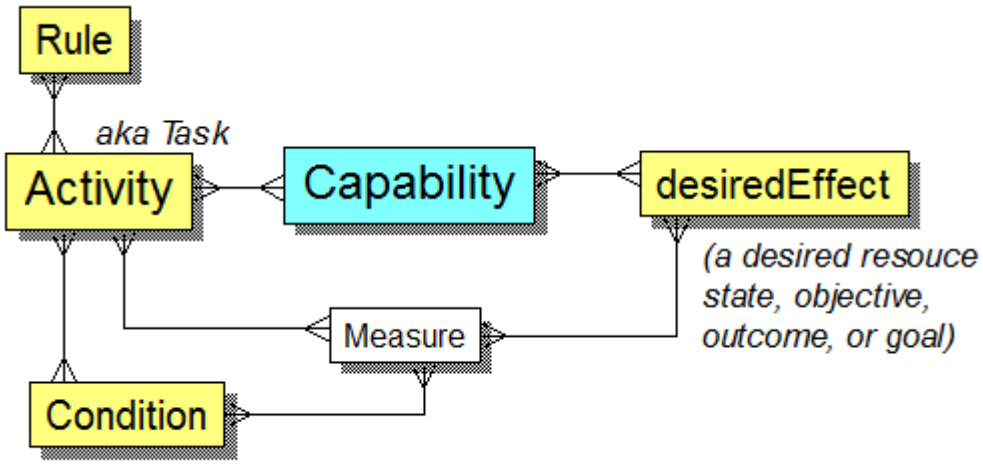
Resource Flow

Reification of Capabilities

- DoDAF definition of Capability
  - "The ability to achieve a Desired Effect under specified [performance] standards and conditions through combinations of ways and means [rules, activities, and resources] to perform a set of activities."

⇒ Capabilities are structures

⇒ to reify them means to reify the structure



For example, in a CV-2, "Capability-1.1 reifies Capability-1" means it reifies the Capability-1 structure:

Capability-1
- •desiredEffects-C1.i
- •Tasks-C1.j
- •Rules-C1.k
- •Conditions-C1.l
- •Measures-C1.m

Where "reify" means:
- •superSubtype,
- •wholePart, or
- •overlap

Capability

For each relationship, use the validation rules for that relationship type

Parts of Capabilities	Description of Relationships	Relationship Type
DesiredEffect	Desired Resource States of Capabilities	WholePartType
Activities	Activities that are parts of Capabilities	WholePartType
Conditions	Conditions under which Activities are performed	OverlapType
Rules	Rules contraining Activities	OverlapType
Measures	Measures pertaining to Activities	SuperSubtype
	Measures pertaining to Conditions	SuperSubtype
	Measures pertaining to Resources	SuperSubtype
	Measures of desire	SuperSubtype
	Measures of effect	SuperSubtype
Performers	Performers performing Activities	OverlapType
	Perfomers desiring the Resource States of Capabilities	OverlapType
	Capabilities of Performers	SuperSubtype
CapabilityTypes	Activities that map to types of Capabilities	TypeInstance

Predicate Calculus of Capability Traceability – Desired Effects

Consistency

$$\text{If } \text{Capability}_x \subseteq \text{Capability}_y, \text{ then } \underset{i}{\forall} \mathit{DesiredEffect}_i \underset{WP}{\subseteq} \mathit{Capability}_{x,\xi} \in \text{Capability}_x,$$

$$\exists \mathit{DesiredEffect}_j \underset{WP}{\subseteq} \mathit{Capability}_{y,\psi} \in \text{Capability}_y \ni \mathit{DesiredEffect}_i \underset{WP}{\subseteq} \mathit{DesiredEffect}_j \wedge$$

$$\left(\begin{array}{l} \underset{\eta}{\forall} \text{measureOfDesiredEffect}_{i,\eta} \ni \mathit{DesiredEffect}_i \in \text{measureOfDesiredEffect}_{i,\eta} \\ \exists\ \text{measureOfDesiredEffect}_{j,\iota} \ni \mathit{DesiredEffect}_j \in \text{measureOfDesiredEffect}_{j,\iota} \wedge \\ \text{measureOfDesiredEffect}_{i,\eta} \subseteq \text{measureOfDesiredEffect}_{j,\iota} \end{array} \right)$$

Completeness

$$\text{If } \text{Capability}_y \subseteq \underset{x}{\cup} \text{Capability}_x, \text{ where for each x, } \text{Capability}_x \subseteq \text{Capability}_y,$$

$$\text{then } \overset{WP}{\underset{j}{\cup}} \mathit{DesiredEffect}_j \in \text{Capability}_y \underset{WP}{\subseteq} \overset{WP}{\underset{x,i}{\cup}} \mathit{DesiredEffect}_{x,i} \in \underset{x}{\cup} \text{Capability}_x \quad \wedge$$

$$\left(\begin{array}{l} \underset{\iota}{\cup} \text{measureOfDesiredEffect}_{j,\iota} \ni \mathit{DesiredEffect}_j \in \text{measureOfDesiredEffect}_{j,\iota} \\ \subseteq \underset{\eta}{\cup} \text{measureOfDesiredEffect}_{x,i,\eta} \ni \mathit{DesiredEffect}_{x,i} \in \text{measureOfDesiredEffect}_{x,i,\eta} \end{array} \right)$$

Predicate Calculus of Capability Traceability – Tasks & Conditions

Consistency

$$\text{If Capability}_x \subseteq \text{Capability}_y, \text{ then } \underset{i}{\forall} Task_i \in \text{Capability}_x,$$

$$\exists Task_j \in \text{Capability}_y \ni Task_i \underset{WP}{\subseteq} Task_j \wedge$$

$$\left(\underset{\eta}{\forall} \text{measureOfTask}_{i,\eta} \ni Task_i \in \text{measureOfTask}_{i,\eta} \; \exists \; \text{measureOfTask}_{j,i} \ni Task_j \in \text{measureOfTask}_{j,\iota} \wedge \right.$$

$$\left. \text{measureOfTask}_{i,\eta} \subseteq \text{measureOfTask}_{j,\iota} \right) \wedge$$

$$\left(\forall Condition_m \underset{WP}{\subseteq} Task_i, \exists Condition_n \underset{WP}{\subseteq} Task_j \ni \right.$$

$$\left. Condition_m \underset{WP}{\subseteq} Condition_n \right)$$

Completeness

$$\text{If Capability}_y \subseteq \underset{x}{\text{U}} \text{Capability}_x, \text{ where for each x, Capability}_x \subseteq \text{Capability}_y,$$

$$\text{then } \overset{WP}{\underset{j}{\text{U}}} Task_j \in \text{Capability}_y \underset{WP}{\subseteq} \overset{WP}{\underset{x,i}{\text{U}}} Task_{x,i} \in \underset{x}{\text{U}} \text{Capability}_x \wedge$$

$$\left(\underset{\iota}{\text{U}} \text{measureOfTask}_{j,\iota} \ni Task_j \in \text{measureOfTask}_{j,\iota} \right.$$

$$\left. \subseteq \underset{\eta}{\text{U}} \text{measureOfTask}_{x,i,\eta} \ni Task_{x,i} \in \text{measureOfTask}_{x,i,\eta} \right) \wedge$$

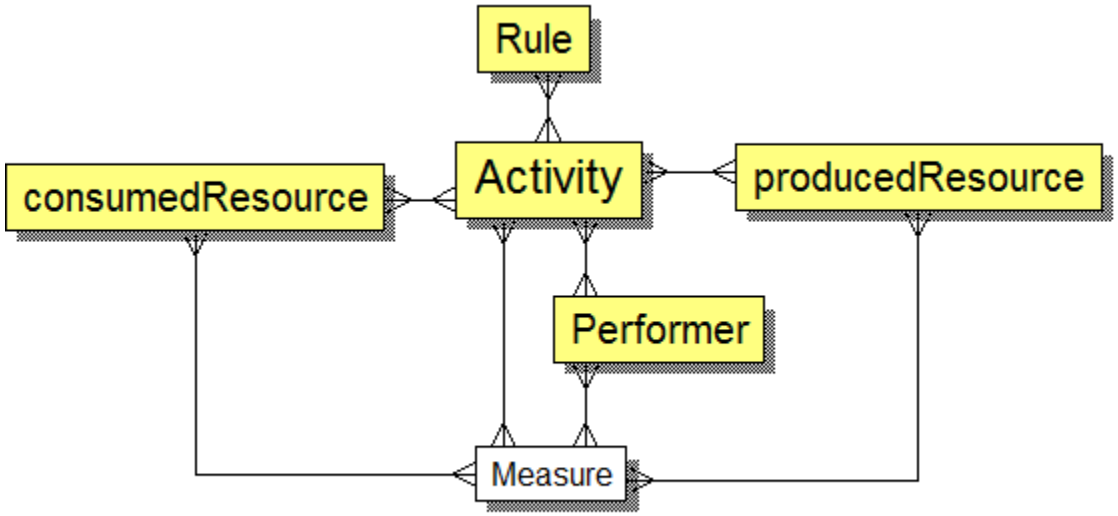
$$\left(\overset{WP}{\underset{m}{\text{U}}} Condition_m \underset{WP}{\subseteq} Task_j \underset{WP}{\subseteq} \overset{WP}{\underset{x,i}{\text{U}}} Condition_n \underset{WP}{\subseteq} Task_i \right)$$

Reification of Activities

- DoDAF definition of Activity –
  - "Work, not specific to a single organization, weapon system or individual that transforms inputs (Resources) into outputs (Resources) or changes their state."

⇒ Activities are structures

⇒ to reify them means to reify the structure



For example, in an OV-5a, "Activity-1.1 reifies Activity-1", means it reifies Activity-1's structure:

- •Activity-1
- •consumedResources-A1.i
- •producedResources-A1.j
- •Rules-A1.k
- •Performers-A1.l
- •Measures-A1.m

Where "reify" means:

- •superSubtype,
- •wholePart, or
- •overlap

Activity

For each relationship, use the validation rules for that relationship type

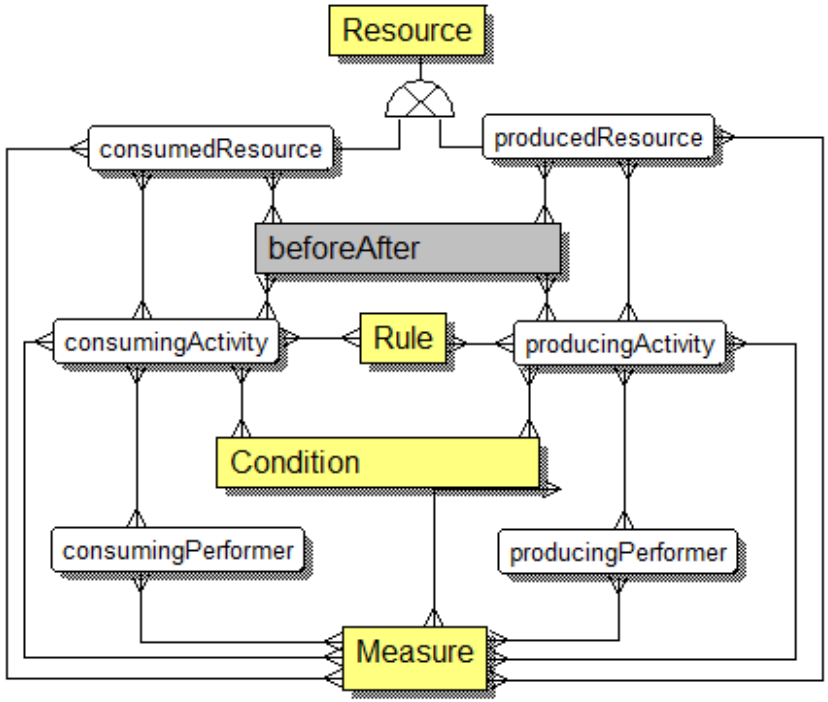
Parts of Activities	Description of Relationships	Relationship Type
Rules	Rules constraining Activities	OverlapType
Conditions	Conditions under which Activities are performed	OverlapType
Performers	Performers performing Activities	OverlapType
Resources	Resources consumed by Activities	OverlapType
	Resources produced by Activities	OverlapType
Measures	Measures pertaining to Activities	SuperSubtype
	Measures pertaining to Conditions	SuperSubtype
	Measures pertaining to Resources	SuperSubtype

Reification of Resource Flow

- DoDAF definition of Resource Flow
  - "The behavioral and structural representation of the interactions between activities (which are performed by performers) that is both temporal and results in the flow or exchange of things such as information, data, materiel, and performers..."

⇒ in an SV-1, each element of a reified resource flow must be a reification of elements from ordinate resource flows

- Note: more complex reifying across allocation levels (e.g., OV→SV) because of typical many-many allocations
  - Some flows get rolled-up
  - New ones get created



Resource Flow

For each relationship, use the validation rules for that relationship type

Parts of Resource Flows	Description of Relationships	Relationship Type
Activities	Sequence of Activities	BeforeAfterType
Rules	Rules contraining Activities	OverlapType
Conditions	Conditions under which Activities are performed	OverlapType
Performers	Performers performing Activities	OverlapType
Resources	Resources consumed by Activities	OverlapType
	Resources produced by Activities	OverlapType
Measures	Measure pertaining to Activities	SuperSubtype
	Measures pertaining to Conditions	SuperSubtype
	Measures pertaining to Resources	SuperSubtype
LocationTypes	Location types of where Resources are	OverlapType

Summary

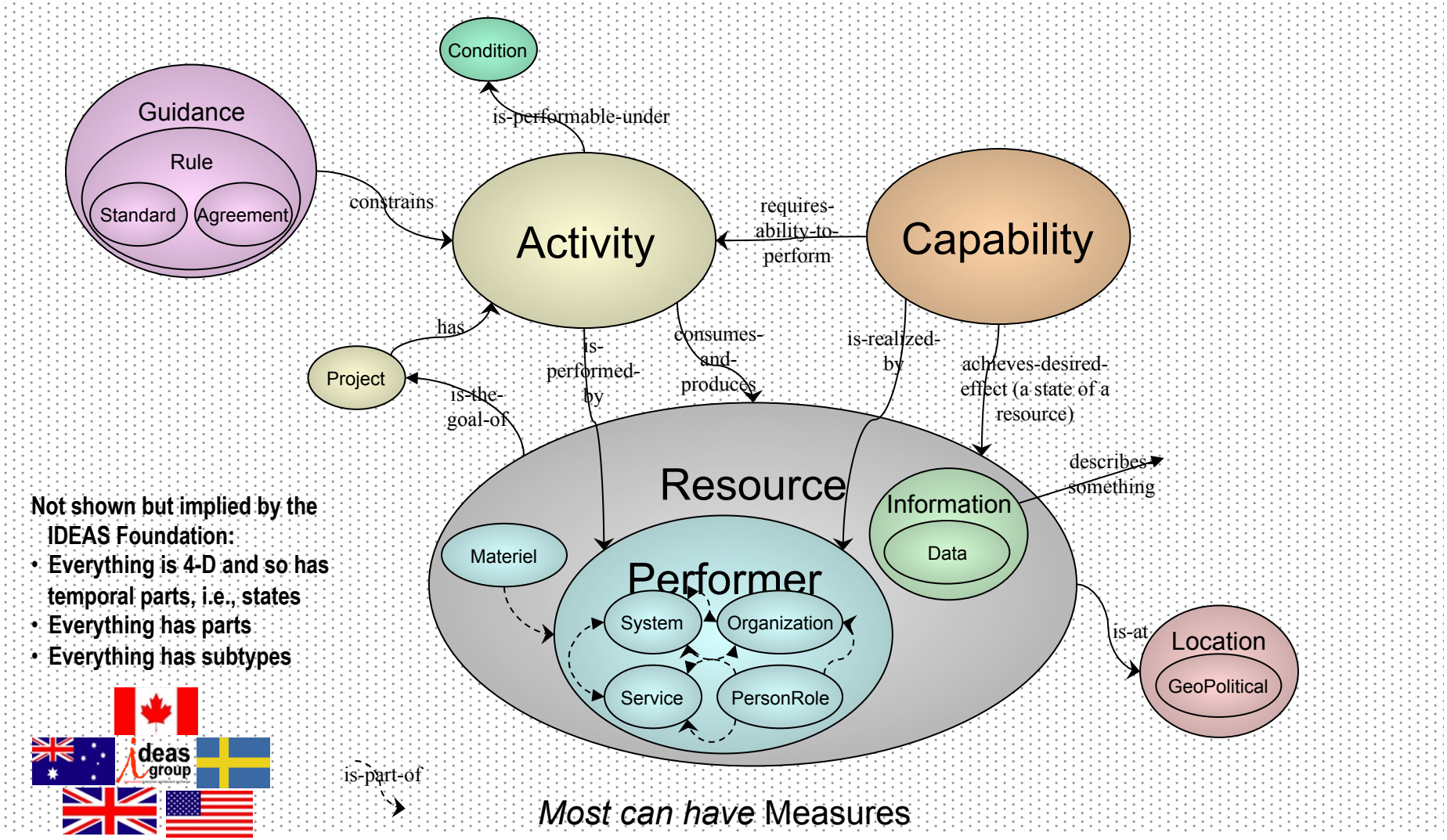
- IDEAS provides a structure for mathematical traceability versus document-based or term-based traceability
- The predicate calculus can be defined
- Future work:
  - How this could be implemented in tools and databases
  - Is traceability between reification levels analogous to homeomorphisms in topology?
  - Determine benefits of mereology versus point set topology and higher-order logic

Questions?

Thank you for your attention!

Backups

How US DoD Thinks of Architectural Descriptions



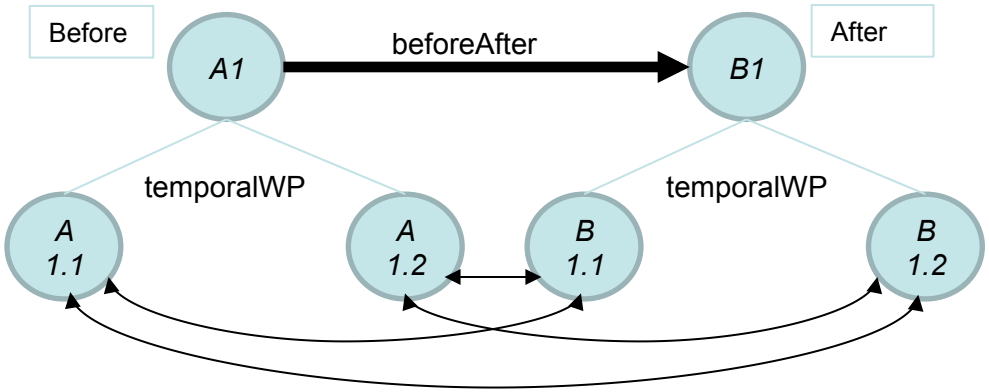
Before-after Notation

A spatio-temporal extent X is a before spatio-extent Y, denoted $X \overset{BA}{\leq} Y$ iff $\forall$ time t_x within X and $\forall t_y$ within $Y, t_x \leq t_y$

1. A proper before-after is denoted $X \overset{BA}{<} Y$ meaning $\forall$ time t_x within X and $\forall t_y$ within $Y, t_x < t_y$
2. $\overset{BA}{\leq}$ and $\overset{BA}{<}$ are transitive: $X \overset{BA}{\leq} Y \wedge Y \overset{BA}{\leq} Z \Rightarrow X \overset{BA}{\leq} Z$
3. $\overset{BA}{<}$ is antisymetric so $X \overset{BA}{<} Y \Rightarrow \neg Y \overset{BA}{\leq} X \wedge X \overset{BA}{\leq} Y \wedge Y \overset{BA}{\leq} X \Leftrightarrow X = Y$
4. $\overset{BA}{<}$ is irreflexive $\neg X \overset{BA}{<} X$
5. $X \overset{BA}{\leq} Y \wedge Y \overset{BA}{\leq} X \Rightarrow t$ is constant for all X and Y

For two sets of spatio-temporal extents $\mathbf{X}$ and $\mathbf{Y}$, $\mathbf{X}=\{X_i\}, \mathbf{Y}=\{Y_j\}$ before-after at the type level is denoted $\mathbf{X} \overset{\mathbf{BAT}}{\leq} \mathbf{Y}$ iff $\forall X_i \in \mathbf{X}\ \exists\ Y_j \in \mathbf{Y} \ni X_i \overset{BA}{\leq} Y_j$ and $\forall Y_j \in \mathbf{Y}\ \exists\ X_i \in \mathbf{X} \ni X_i \overset{BA}{\leq} Y_j$

Validation for Relationship Type: beforeAfter

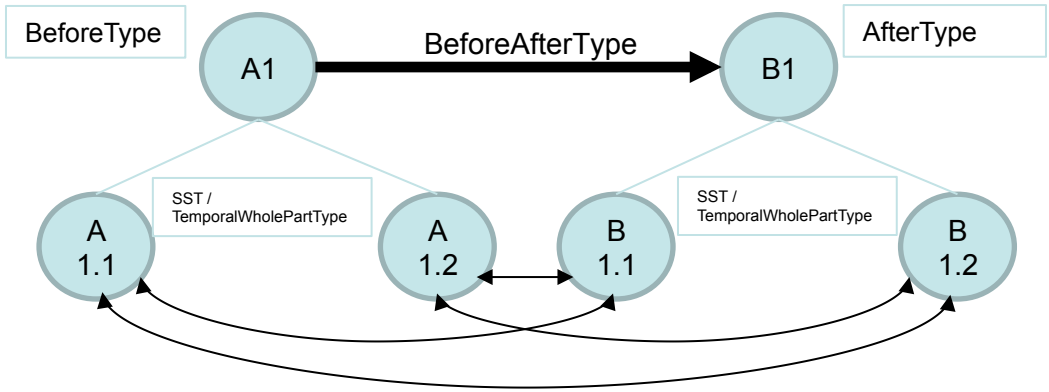


Validation Check:

1. Every temporalWP of *A1* has a beforeAfter with every temporalWP of *B1*

2. If all the temporalWPs of *A1* have a beforeAfter with every temporalWP of *B1*, then *A1* must be before *B1*

Validation for Relationship Type: BeforeAfterType



Validation Check:

1. Every temporalWPT of A1 must have a BeforeAfterType with B1 and conversely, every temporalWPT of B1 has an inverse BeforeAfterType with A1

2. Every SST of A1 must have a BeforeAfterType with B1 and conversely, every SST of B1 has an inverse BeforeAfterType with A1

3. If all the SST of A1 have a BeforeAfterType with any SST of B1, and all the SST of B1 have an inverse BeforeAfterType with any SST of A1, then A1 must be a BeforeAfterType of B1 (Note 1)

4. If all the temporalWPT of A1 have a BeforeAfterType with every temporalWPT of B1 and all the temporalWPT of B1 have an inverse BeforeAfterType with every temporalWPT of A1, then A1 must be a BeforeAfterType of B1 (Note 2)

Note 1: SST assumes complete subsets
Note 2: temporal WPT assumes complete parts

Predicate Calculus of Activity Traceability – Rules

Consistency

If $\text{Activity}_x \subseteq \text{Activity}_y$, then $\forall_i Rule_i \underset{OV}{\mathrm{I}} Activity_{x,\xi} \in \text{Activity}_x$,

$\exists \overset{WP}{\underset{j}{\mathrm{U}}} Rule_{y,\psi,j} \ni Rule_{y,\psi,j} \underset{OV}{\mathrm{I}} Activity_{y,\psi} \in \text{Activity}_y \wedge Activity_{y,\psi} \in \text{Activity}_x \wedge Rule_i \underset{WP}{\subseteq} \overset{WP}{\underset{j}{\mathrm{U}}} Rule_{y,\psi,j}$

If $\text{Activity}_x \underset{WPT}{\subseteq} \text{Activity}_y$, then $\forall_i Rule_i \underset{OV}{\mathrm{I}} Activity_{x,\xi} \in \text{Activity}_x$,

$\exists \overset{WP}{\underset{j}{\mathrm{U}}} Rule_{y,\psi,j} \ni Rule_{y,\psi,j} \underset{OV}{\mathrm{I}} Activity_{y,\psi} \in \text{Activity}_y \wedge Activity_{x,\xi} \underset{WP}{\subseteq} Activity_{y,\psi} \wedge Rule_i \underset{WP}{\subseteq} \overset{WP}{\underset{j}{\mathrm{U}}} Rule_{y,\psi,j}$

Completeness

If $\text{Activity}_y \subseteq \underset{x}{\mathrm{U}} \text{Activity}_x$, where for each x, $\text{Activity}_x \subseteq \text{Activity}_y$,

then $\forall Activity_{y,\psi} \in \text{Activity}_y, \overset{WP}{\underset{j}{\mathrm{U}}} Rule_{y,\psi,j} \underset{OV}{\mathrm{I}} Activity_{y,\psi} \underset{WP}{\subseteq} \overset{WP}{\underset{x,\xi,i}{\mathrm{U}}} Rule_{x,\xi,i}$

where $Rule_{x,\xi,i} \underset{OV}{\mathrm{I}} Activity_{x,\xi} \in \text{Activity}_x$

If $\text{Activity}_y \underset{WPT}{\subseteq} \overset{WPT}{\underset{x}{\mathrm{U}}} \text{Activity}_x$, where for each x, $\text{Activity}_x \underset{WPT}{\subseteq} \text{Activity}_y$,

then $\forall Activity_{y,\psi} \in \text{Activity}_y, \overset{WP}{\underset{j}{\mathrm{U}}} Rule_{y,\psi,j} \underset{OV}{\mathrm{I}} Activity_{y,\psi} \underset{WP}{\subseteq} \overset{WP}{\underset{x,\xi,i}{\mathrm{U}}} Rule_{x,\xi,i}$

where $Rule_{x,\xi,i} \underset{OV}{\mathrm{I}} Activity_{x,\xi} \in \text{Activity}_x$