Important Points

- TRAK isn't UML - TRAK views declare statements/sentences (tuples /triples) e.g. 'Organisation. IET *has part* Organisation. Council'
- every TRAK architecture description element has attributes to hold information.
- UML, BPMN etc are architecture description languages & may be used to represent TRAK views
- TRAK is specified in a solution- / notation-agnostic way using 3 documents:-
  - TRAK00004. TRAK
  - TRAK00001. TRAK. Viewpoints
  - TRAK00002. TRAK. Metamodel

Minimal Process

- establish architecture task stakeholder's concerns
- select views needed to match these concerns using the TRAK viewpoints
- use MV-02 to capture concerns, outline models needed/developed and record findings/analysis and document the Architecture Description so that it can be understood and re-used

Constraints

- each view must conform to its viewpoint (TRAK Viewpoints)
- views must meet consistency rules for the collection of views
- TRAK Bye Laws include:
  - no orphans - minimum allowed is an architecture description tuple/ triple.
  - every relationship must be visible in at least one view
- colours of each element must conform to those used in the metamodel (specified in Overall TRAK specification) - a graphic may be used instead

Select the TRAK Viewpoints That Address the Concerns Raised

The TRAK viewpoints and the typical or generalised concerns they address are:

- **EVp-01. Enterprise Goal**. 'What is the Enterprise and what goals does it set out to achieve?
- **EVp-02. Capability Hierarchy**. 'What are the enduring capabilities the enterprise requires and how is capability measured?'
- **EVp-03. Capability Phasing**. 'How is capability delivered over time? Are there any gaps?'
- **CVp-01. Concept Need**. 'Have the concept needs been identified?'
- **CVp-03. Concept Item Exchange**. 'Have the items exchanged by concept nodes been identified? What is required to satisfy the concept needs?'
- **CVp-04. Concept Activity to Capability Mapping**. 'How/are concept activities sufficient to deliver capability?'
- **CVp-05. Concept Activity** 'What does each concept node need to do?'
- **CVp-06. Concept Sequence**. 'How are concept activities ordered? Is it important?'
- **PrVp-01. Procurement Structure**. 'What is the project structure? How is it governed?"
- **PrVp-02. Procurement Timeline**. 'What other projects is this dependent on? How does their delivery time affect us?"
- **PrVp-03. Procurement Responsibility**. 'What responsibilities do organisations or jobs have in relation to a project or time? Are their boundaries clear?'
- **SVp-01. Solution Structure**. 'What does the solution consist of? Is it structured sensibly? What is the organisation structure / membership? How does responsibility (scope/jurisdiction) apply to the solution components?'
- **SVp-02. Solution Resource Interaction**. 'How are resources connected together? How are the organisations, jobs & roles connected? Have the interactions/ interfaces/exchanges been characterised?'
- **SVp-03. Solution Resource Interaction to Function Mapping**. 'Are there interactions/ interfaces that cannot be justified by functional need? Do we have functions that cannot be realised because there isn't an interchange?'
- **SVp-04. Solution Function**. 'Have all solution functions been identified? What does each part do?'
- **SVp-05. Solution Function to Concept Activity Mapping**. 'Do the solution functions meet all of the concept activities? Is there unwanted solution functionality?'
- **SVp-06. Solution Competence**. 'Does the organisation or job through its role have the necessary competence to conduct the function? Is the competence consistent with the solution?'
- **SVp-07. Solution Sequence**. 'In what order do things need to happen?'
- **SVp-11 Solution Event Causes.** 'How robust is the system to unwanted events? How dependable is the system? What causes (feared) events?
- **SVp-13 Solution Risk**. What threats is the system of interest exposed to? What are the vulnerabilities of the system of interest? What

are the risks posed to the system, or to a third party by the system? How does the solution design mitigate or address the vulnerabilities, threats and risks?

- **MVp-01. Architecture Description Dictionary.** 'Is the architecture portable? Can it be understood in the way it was intended to be?'
- **MVp-02. Architecture Description Design Record**. 'Do we understand the scope of the architectural task? What are the issues and findings that resulted?'

- **MVp-03. Requirements & Standards**. 'Have all the constraints been identified? What constraints/ requirements through normative documents/standards apply (or will apply) to the system, project, enterprise?'
- **MVp-04. Assurance**. 'What are the claims made? What is the basis for the claim? Is there any evidence?'

TRAK viewpoints have a 'p' in the identifier whereas the conforming view doesn't e.g. MVp-02 (the **v**iew**p**oint) vs MV-02 (the **v**iew)

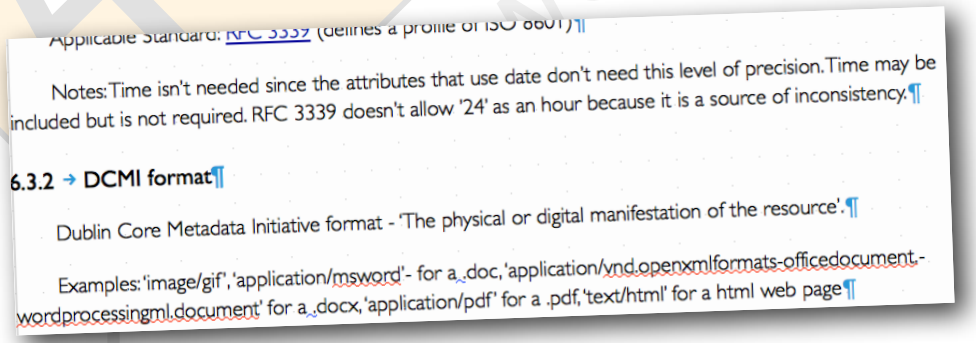
TRAK Metamodel

- Defines the elements that appear in TRAK views (architecture description elements) and how they can be connected together.
- Defines meaning of each element and its attributes.

ıte¶	Element Type Name¶	Perspective¶	Definition¶	Attributes¶	Tests For¶
an,	Protocol¶	Solution¶	A set of rules governing the exchange procedure between things.¶	+ Architecture Description Element¶	¶
pt,	Requirement¶	Management¶	An atomic requirement or constraint - a	+ Architecture Description Element¶	atomic¶

Implementation of TRAK

- Implementation of the 3 TRAK specification documents is controlled by an implementation specification
  - TRAK00005. TRAK. Implementation.Architecture Description Elements.
- specifies
  - naming, applicable standards
  - TRAK can be implemented using any appropriate notation to describe the tuples
    - text
    - UML, SysML
    - directed property graph
    - semantic notation e.g. RDF
      - the choice depends on
      - tools available
      - experience familiarity
      - use of attributes
      - use / exploitation of relationships / tuples once described e.g. ad hoc query.
  - expected life / storage of architecture description



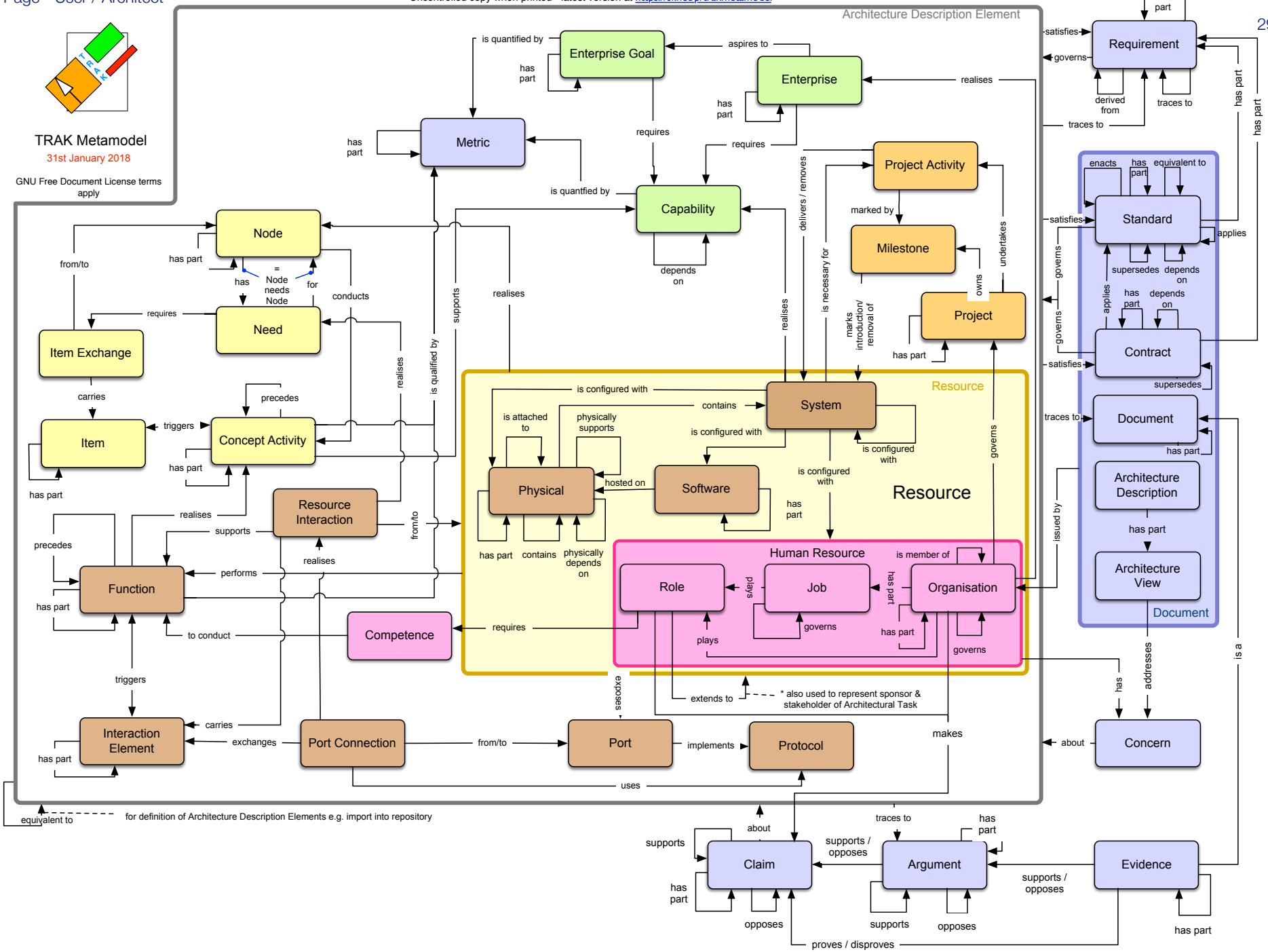
Applicable Standard: RFC 3339 (defines a profile of ISO 8601)¶

Notes: Time isn't needed since the attributes that use date don't need this level of precision. Time may be included but is not required. RFC 3339 doesn't allow '24' as an hour because it is a source of inconsistency.¶

6.3.2 → DCMI format¶

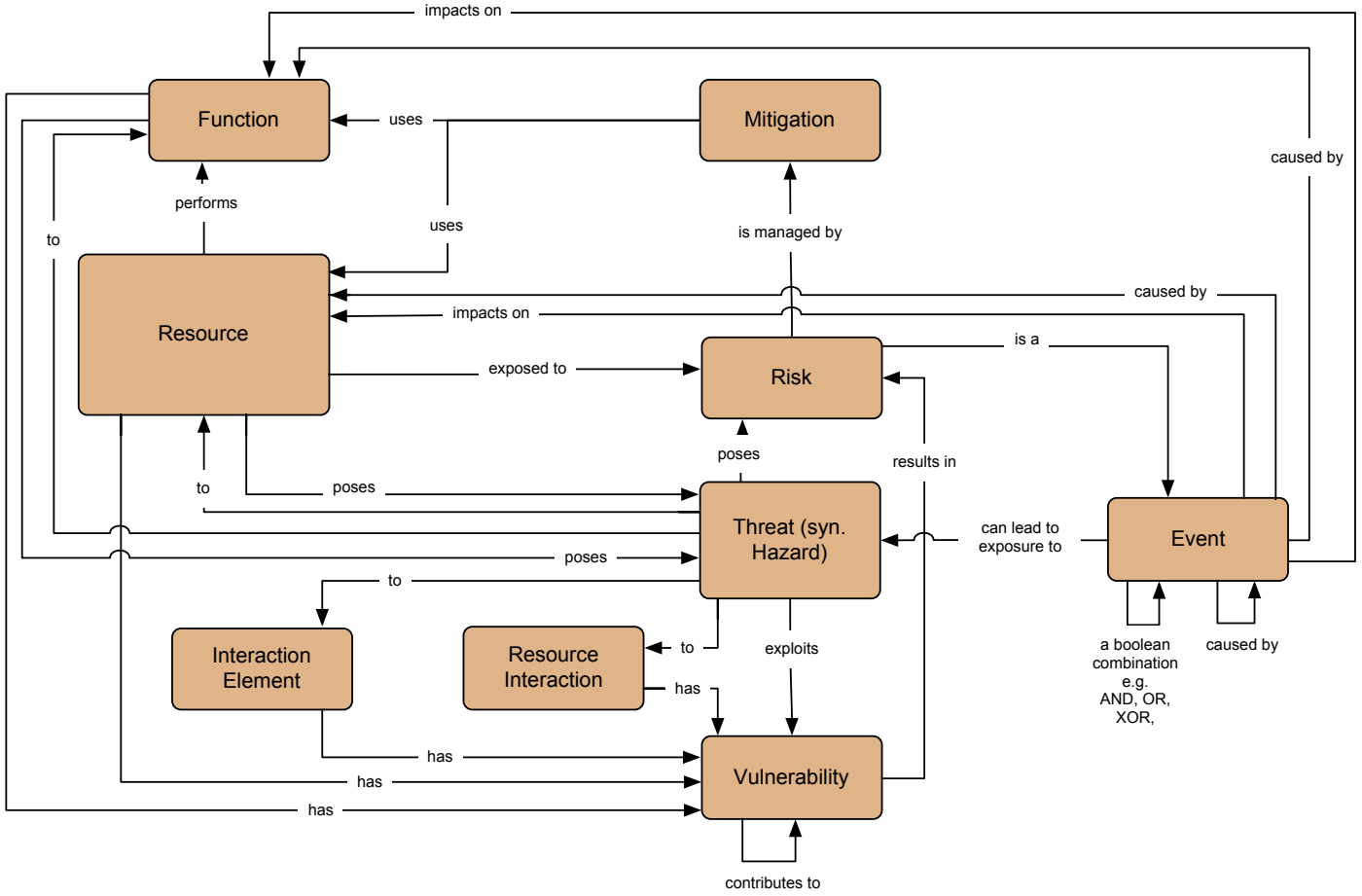
Dublin Core Metadata Initiative format - 'The physical or digital manifestation of the resource'.¶

Examples: 'image/gif', 'application/msword' - for a .doc, 'application/vnd.openxmlformats-officedocument.wordprocessingml.document' for a .docx, 'application/pdf' for a .pdf, 'text/html' for a html web page¶


TRAK Metamodel
31st January 2018
GNU Free Document License terms apply
Architecture Description Element
Enterprise Goal
Enterprise
Metric
Capability
Project Activity
Milestone
Project
Node
Need
Item Exchange
Item
Concept Activity
Resource Interaction
Function
Competence
Interaction Element
Port Connection
Port
Protocol
Resource
System
Physical
Software
Human Resource
Role
Job
Organisation
Requirement
Standard
Contract
Document
Architecture Description
Architecture View
Concern
Claim
Argument
Evidence
is quantified by
aspires to
has part
realises
requires
delivers / removes
is necessary for
marked by
undertakes
owns
marks introduction/ removal of
from/to
has
= Node needs Node
for
conducts
supports
is qualified by
carries
triggers
precedes
performs
to conduct
is configured with
contains
is attached to
physically supports
hosted on
has part
physically depends on
plays
governs
issued by
satisfies
derived from
traces to
enacts
equivalent to
applies
supersedes
depends on
addresses
is a
exposes
extends to
* also used to represent sponsor & stakeholder of Architectural Task
makes
about
implements
uses
equivalent to
for definition of Architecture Description Elements e.g. import into repository
supports / opposes
proves / disproves

TRAK Metamodel

31st January 2018



References

- TRAK00004. TRAK. sf.net/projects/trak
- TRAK00002. TRAK Metamodel sf.net/projects/trakmetamodel
- TRAK00001. TRAK Viewpoints sf.net/projects/trakmetamodel
- TRAK00005. TRAK. Implementation. Architecture Description Elements. https://sourceforge.net/projects/trak/files/Implement%20TRAK/
- TRAK Community trak-community.org
- TRAK Implementations trak.sf.net/implementations.html
- Twitter - TRAK_AF