TRAK

ENTERPRISE ARCHITECTURE FRAMEWORK

VIEWPOINTS

COPYRIGHT

CONTENTS

GNU FREE DOCUMENTATION LICENSE

GNU Free Documentation License Version 1.3, 3 November 2008

The text of the license is at http://www.gnu.org/licenses/fdl-1.3.html (Error: Reference source not found)

WARRANTY DISCLAIMERS

This Document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NETWORK LOCATION

This document is available at http://sf.net/p/trakviewpoints

HISTORY

Changes to the TRAK viewpoints are also tracked via a RSS feed . and in the trakviewpoints project version-controlled repository at http://sourceforge.net/p/trakviewpoints/code/

Author(s)	Date	Changes
Nic Plum	20th August 2019	Added new Figure 3-5 Architecture View Set Dependencies (Graph). #57 - CVp-05 Concept Activity. Removed 'Item *has part* Item' from optional tuples. #58 - CVp-01 Concept Need. Added '*for* Node' added to 'Node *has* Need' as alternative to 'Node *needs* Node'. #59 - SVp-11 Solution Event Causes. Replaced 'Event c*an lead to exposure to* Risk' with 'Event c*an lead to exposure to* Threat'. #60 - SVp-13 Solution Risk. Replaced 'Event c*an lead to exposure to* Risk' with 'Event c*an lead to exposure to* Threat'. #61 - Replaced Figure 3-4 - Master Architecture Views – Architecture Task Description Elements with correct figure (was showing assurance elements). #62 - SV-02 supplies Interaction Element to SV-04 removed from Figure 3-1 Master Architecture Views Create View Dependencies. #63 - 'Argument' and 'Evidence' added to Figure 3-3 - Master Architecture Views – Assurance Elements. #64 - 'Contract' added to Figure 3-2 Master Architecture Views – Contracts, Requirements & Standards.
Nic Plum	8th February 2018	#56. MVp-02 Architecture Description Design Record. Restored viewpoint concerns addressed.
Nic Plum	31st January 2018	SVp-01 Solution Structure. Added 'Physical *physically supports* Physical', 'Physical *is attached to* physical'
Nic Plum	11th December 2017	55. Example PrVp-02 corrected ('*marks introduction of'* corrected '*to marks removal of'*)

Author(s)	Date	Changes
		40. Added Document to MV-02 on Figure 3-4. 39. Added Contract to Figure 3-2 Master Architecture Views – Contracts, Requirements & Standards Figure 3-1 Master Architecture Views Create View Dependencies 38. Added CV-01 supplies Need to CV-03, SV-02 supplies Resource to SV-03 37 Added SV-01 supplies Resource to SV-04
Nic Plum	8th December 2017	Metric 'declared' on respective EV-03, CV-05 or SV-04 view in Table 3-1 Master Architecture View for Each TRAK Metamodel Element. **General**. Replaced 'stereotype' with 'metamodel element'. All Viewpoints Changed stakeholder concerns figure into table. Deleted 'Mandatory Tuples' figures. Changed 'Mandatory Tuples' section title name to 'Declared Tuples' Tuples forming Well-formedness criteria stated where not already explicit. **Specific** CVp-03. Added a Well-formedness constraint on the direction of the Item Exchange. PrVp-01 Added a Well-formedness constraint recognising Project has part Project. **Impact of Metamodel Changes** All Viewpoints 'Architecture Description Element *satisfies* Requirement', 'Architecture Description Element *satisfies* Contract', 'Architecture Description Element *satisfies* Standard', 'Contract *governs* Architecture Description Element', 'Architecture Description Element *traces to* Contract' tuples added to Universal Tuples. SVp-13 Solution Risk. Added 'Threat (syn. Hazard) *to* Function', 'Function *poses* Threat (syn. Hazard)'

Author(s)	Date	Changes
		MVp-01 Architecture Description Dictionary. Added 'Architecture Description Element *equivalent to* Architecture Description Element, Architecture Description Element *is a* Architecture Description Element MVp-02 Architecture Description Design Record. Added 'Document *has part* Document'. MVp-03 Requirements & Standards. Addition of 'Contract *governs* Architecture Description Element', 'Architecture Description Element *satisfies* Requirement / Standard / Contract', 'Architecture Description Element *traces to* Contract', 'Requirement *derived from* Requirement', 'Contract *issued by* Organisation', **Bugs** 54. SVp-01 well-formedness criteria for Role Extent form widened to include all Resources not just System. 53. User (of Concept) stakeholder removed from PrVp-02 Procurement Timeline viewpoint. 52. MV-01 Architecture Description Dictionary View added to all view sets in Table 3-2 Minimum Allowed View Sets. 51. PrVp-03 Procurement Responsibility Viewpoint – User (of Concept) stakeholder removed. 50. MVp-03 Requirements & Standards Viewpoint Concerns changed to be more generic wrt application. Also affects Table 2-1. 49. MVp-02 Mandatory Tuples – 'Standard *governs* Architecture Product' moved from Architecture Task Findings to ISO 42010 AD Scope section. 48. SVp-07 Solution Sequence – changed Well-formedness criteria for Function Sequence form. 47. Added missing 'Architecture Task *delivers* Architecture Product (Architecture Description, Architecture View)' to MVp-02 - Mandatory Tuples - ISO 42010 AD Scope 46 Added missing 'Architecture Description Element *traces to* Standard' to MVp-02 Optional Tuples

Author(s)	Date	Changes
		45. Added missing 'Organisation or Job *has* Concern' to MVp-02 – Mandatory Tuples - ISO 42010 AD Scope 44. Added missing 'Architecture Task *has part* Architecture Task' to MVp-02 – Mandatory Tuples - ISO 42010 AD Scope. 43. Added missing Architecture Task delivers Architecture Product (Architecture Description, Architecture View) to MVp-02 – Mandatory Tuples - ISO 42010 AD Scope 42. Changed 'architecture' to 'architecture description' in concern for MVp-01 41. Added Contract to Well-formedness criteria for MVp-03 **Requests** 26. MVp-01 Architecture Description Dictionary – added Architecture Description element *is a* Architecture Description Element, Architecture Description Element *equivalent to* Architecture Description Element 25. MVp-03 Requirements & Standards Viewpoint made more compact – removed sub sections showing constraints in each separate perspective. 24. (see Bug 44)
Nic Plum	23rd January 2016	SVp-11 Solution Event Causes. Optional Tuples added 'System *is configured with* Resource', 'Software *hosted on* Physical', 'Physical *contains* System', 'Physical *has part* Physical', 'Software *has part* Software', 'Organisation *is member of* Organisation', 'Organisation *has part* Organisation', 'Organisation *has part* Job, 'Job *plays* Role, 'Organisation *plays* Role', 'Resource *performs* Function'. SVp-13 Solution Risk. Declared Tuples - added 'Vulnerability *contributes to* Vulnerability'. Changed 'Risk *is managed using* Mitigation' to 'Risk *is managed by* Mitigation'. Changed 'Event *causes* Event' to 'Event *caused by* Event'. Optional Tuples Added - Context – Containing

Author(s)	Date	Changes
		System – 'System *is configured with* Resource', 'Software *hosted on* Physical', 'Physical *contains* System', 'Physical *has part* Physical', 'Software *has part* Software', 'Organisation i*s member of* Organisation', 'Organisation *has part* Organisation', 'Organisation *has part* Job', 'Job *plays* Role', 'Organisation *plays* Role'. Well-Formedness - Identification – added 'Threat *to* Function' and 'Resource *performs* Threat', 'Threat *to* Resource Interaction', 'Resource Interaction f*rom* / *to* Resource', 'Threat *to* Interaction Element' , 'Resource *exposes* Port' and 'Port Connection *to* / *from* Port'. #36 - Management & Control - added 'Mitigation *uses* Resource' , 'Mitigation *uses* Function', 'Resource *performs* Function' #35. MVp-02 Architecture Description Design Record - Well-Formedness - Declared Tuples - ISO 42010 AD Scope - changed 'exactly one Architecture Task' to 'at least one Architecture Task'
Nic Plum	1st January 2016	Changed '2013' to '2016' in COPYRIGHT. Corrected reference to GFDL in GNU Free Documentation License. Modified Figure 1-1-Context for the TRAK Architecture Viewpoints Document (This Document). Modified Figure 1-2-Normative TRAK Documents - Logical Definitions vs Implementation of TRAK. Added 2 viewpoints – SVp-11 Solution Event Causes and SVp-13 Solution Risk. Added SVp-11 and SVp-13 concerns to Table 2-1 - TRAK Viewpoints - Concerns Addressed Added Event, Mitigation, Risk, Threat and Vulnerability to Table 3-1 Master Architecture View for Each TRAK Metamodel Element. Added minimum viewsets VS27 and VS28 to Table 3-2 Minimum Allowed View Sets Modified Error: Reference source not found, Figure 3-2, Figure 3-3. Added SVp-11 and SVp-13.

Author(s)	Date	Changes
		Added new Figure 3-4 - Master Architecture Views – Architecture Task Description Elements. Added short form of Sourceforge URIs and documents numbers to TRAK documents in References.
Nic Plum	14th June 2015	Conversion to Open Office document format. Added MVp-04 to Table 2-1 (Viewpoint Selection) #34 . Added 'Requirement *governs* Architecture Product' to MVp-02 #33. Added 'Job / Organisation *plays* Role' and 'Role *extends to* Resource' as optional context to MVp-04 Assurance Viewpoint #22. Added index.
Nic Plum	24th December 2014	Added 'Standard *applies* Standard' , 'Standard *governs* Standard', 'Requirement *governs* Architecture Description Element' to MVp-03 Requirements & Standards (metamodel change requests #26, #27 and #28). Requirement *governs* Architecture Description Element' affects 'Universal Tuples' section of every viewpoint. Generic roles of 'auditor' and 'regulator' added to TRAK Architecture Viewpoints - Introduction and to MVp-03 and MVp-04 viewpoints (#20). Added MVp-04 Assurance viewpoint - adds 'Claim *about* Architecture Description Element' to 'Universal Tuples' section of every viewpoint. New Figure 12-3 Master Architecture View - Claim Changed '*necessary for*' to 'i*s necessary for*' in 'System *is necessary for* Project Activity 'in PrVp-02 and PrVp-03 (metamodel change #30)
Nic Plum	5th February 2013	#30 Added requirement to identify version of AD and record rationale to MVp-02 Architecture Description Design Record Viewpoint to address requirements of ISO/IEC/IEEE 42010:2011 5.1 and 5.6. #18 Added 'System realises Capability' to SVp-01 Solution Structure Viewpoint

Author(s)	Date	Changes
Nic Plum	7th December 2012	#27 Added Physical *contains* System and Physical *contains* Physical to SVp-01 Solution Structure. Updated links as a result of change in Sourceforge platform hosting.
Nic Plum	13th April 2012	#3475115 Deleted CVp-02 Concept Viewpoint. #3507818 Added optional context relationships to SVp-07
Nic Plum	2nd Oct 2011	#3387152 Metamodel change 30th Sept. 2011. Added Architecture Description has part Architecture Description to MVp-02. Added Figure 1-2. Updated Figure 12-1 and Figure 12-2
Nic Plum	6th August 2011	#3305946 Added Contract has part Contract to MVp-03
Nic Plum	11th April 2011	#3211371 added missing label to CVp-05 mandatory tuples diagram. #3210825/ #3210840 order-like tuples removed from SVp-04/CVp-05 leaving SVp-07/CVp-06 as focus for addressing order. Consequential change to consistency rules. #3231406 removed Enterprise aspires to Enterprise Goal from EVp-02. PrVp-03 - removed optional tuples Role *requires* Competence, Competence *to conduct* Function
Nic Plum	8th March 2011	Corrected cut and paste error - stakeholder concern diagrams EVp-02, EVp-03. Addition made to well-formedness for CVp-02. Clarified correspondence rule for SVp-07. Added SV-02/SV-07 correspondence rule to SVp-02. Added view set VS26 for SV-07 and modified VS21 to recognise 2 forms of SV-07. Added clarification text to MVp-03.
Nic Plum	2nd March 2011	#3185771 Metric has part Metric added to EVp-02, CVp-05 and SVp-04. #3185817 Function *triggers* Interaction Element added to SVp-03. #3185819 Architecture Description *has part* Architecture View added to MVp-02. #3185839, #3185845 missing mandatory tuples added to

Author(s)	Date	Changes
		MVp-02., MVp-03. #3185820 Organisation has part Organisation added to context for SVp-06. #3185829 Function has part Function, Interaction Element has part Interaction Element added as context for SVp-07. #3190237 addition made to concerns addressed by EVp-03 to recognise solutions realising capability (or not). SVp-03 modified to add Resource Interaction supports Function. Affects tuples and also consistency rules for SVp-07 and note for SVp-02.
Nic Plum	14th Feb 2011	CVp-03, SVp-04 added allowed type for interaction. CVp-03 added Item Exchange *from*/*to* Node (metamodel change #3171404). MVp-03 - added views needed for Management Constraints
Nic Plum	3rd Feb 2011	Added presentation examples to CVp-01, CVp-03, CVp-06, SVp-01, SVp-02, SVp-07, PrVp-03 Modified mandatory tuples - CVp-06, SVp-07. Modified well-formedness rules - SVp-07. #3167755 EVp-01 - removed duplicate tuple. #3167763 Added explanation about taxonomy diagrams to EVp-02.
Nic Plum	27th Jan 2011	trakmetamodel #3138601 Added 'Function *precedes* Function', 'Concept Activity *precedes* Concept Activity' - CVp-05, CVp-06, SVp-04, SVp-06. #3161826 Added 'Requirement *has part* Requirement' - MVp-03 MVp-03 Diagrams added for mandatory tuples various uses for constraints . CVp-04, SVp-05 - replaced example illustrating use of Concern objects. #3165789 SVp-01 adding missing tuples to mandatory tuples, configuration form of viewpoint.
Nic Plum	20th Jan 2011	#3140703 Added missing 'is quantified by Metric' to EVp-02, CVp-05 and SVp-04. Added section on

Author(s)	Date	Changes
		ADL choice. #3140866 Added new section with instructions for identification of non-conforming architecture products in a TRAK-compliant architecture description. Added fragments of TRAK metamodel to viewpoints - EVp-01, EVp-02, EVp-03, CVp-01, CVp-02, CVp-03, CVp-05, CVp-06, PrVp-01, SVp-04, SVp-05, SVp-06, SVp-07 Replaced GNU Free Documentation License text with link to the text. Changed Figure 1-1 and text to recognise any implementation of TRAK. Added Well-Formedness heading to each viewpoint to identify minimum content requirements for views. #3161777 added Competence to solution constraint version of MVp-03 #3161782 added Standard to tuples in constraints versions of MVp-03. Added examples under 'presentation' for viewpoints. #3160753 (in TRAK metamodel) change of name from Architecture Element to Architecture Description Element - affects MVp-01 and MVp-02. Responded to INCOSE UK AWG change requests: #3138698 SV-01 changed to SV-01 in Baseline Comparison with MODAF 1.2. #3142801 Changed EVp-03 concern to 'when' not 'how'. Common or whole-framework sections moved into a new document - 'TRAK Architecture Framework'. trak.sourceforge.net (Important Ideas, Standards Affecting TRAK, Glossary, Choice of ADL, Conformance with TRAK, Architecture Perspectives, Use of Colour, TRAK Bye Laws, Minimal Modelling Process)
Nic Plum	26-July-2010	Viewpoints. - added statement under optional tuples linking to master architecture views needed if elements added. Added short rationale to each required view. Added missing hyperlinks to Minimum Allowed View Sets. Added metamodel fragment diagrams to CVp-04, PrVp-02, PrVp-03,

Author(s)	Date	Changes
		SVp-01, SVp-02, SVp-03. PrVp-03 mandatory tuples redefined (incomplete statement / tuple path previously). Added missing tuple to SVp-02 (#3011459).
Nic Plum	16-July-2010	Added Figure 1 for context (other figure numbers changed). Added Bye Laws BLV-7 and BLV-8 for viewpoint collection design.
Nic Plum	03-Jun-2010	Corrected perspective names to Enterprise and Concept in Table 1 - TRAK Viewpoints - Concerns Addressed (#3011003). Node has part Node moved to mandatory tuples for CV-01 Concept Need Viewpoint (#3010964). Interaction Element has part Interaction Element added to SV-02 Solution Resource Interaction Viewpoint (#3011058). Consistency rule added to PrV-01 and SV-01 to ensure separation of Organisation in models of the business and the solution delivered by the business.
Nic Plum	29-Apr-2010	Baseline wrt MODAF moved to end of document. #2989344 Operational Perspective renamed Concept Perspective - CV prefix - to avoid application to the purely day to day activity. Old OVp-01 to OVp-06 affected. Old OVp-02 name changed to 'Concept View'. #2993201 Capability Perspective renamed Enterprise Perspective - EV prefix to avoid naming conflicts & name after the 'thing' rather than its activity. Old CVp-01 to CVp-03 affected. Affects minimum allowed view sets. 'Operational Activity' metamodel element now 'Concept Activity' - affects old OVp-02, OVp-04, OVp-05, OVp-06, SVp-04, SVp-05.
Nic Plum	01-Apr-2010	OVp-07 Operational Constraints & SVp-10 Solution Constraints incorporated into a single MVp-03 Requirements & Standards viewpoint to allow constraints to be described for any perspective not just two. Figure 5 added. Table added to show master architecture view for each metamodel element.

Author(s)	Date	Changes
Nic Plum	22-Mar-2010	Changed name of OV-01 to Operational Need - again consistency & keying into metamodel. Corrected SV-01 consistency rule wrt role extent.
Nic Plum	19-Mar-2010	Added 'Description' to names for MV-01 and MV-02 to a) make it clear that these relate to the architecture description (model) rather than architecture itself and b) consistency - key viewpoints(views) to metamodel.
Nic Plum	17-Mar-2010	Added a section covering important ideas. Incorporated removal of 'sponsors' Architecture Task and substitution by Role extends to Architecture Task in metamodel. MVp-02 affected.
Nic Plum	05-Mar-2010	CVp-01 - added Organisation realises Enterprise. CVp-02 - added Enterprise requires Capability, Organisation realises Enterprise. OVp-01 - added Resource realises Node. OVp-05 - added Function realises Operational Activity, Operational Activity supports Capability and consistency rule. SVp-01 - added Organisation realises Enterprise. SVp-02 added Resource realises Node and consistency rule. SVp-04 - added Function realises Operational Activity and consistency rule SVp-05 - added consistency rule.
Nic Plum	02-Mar-2010	Added consistency rules to PrVp-02 between Milestone / Project Activity and System. Incorrect concerns addressed by SVp-05, SVp-05 in Viewpoint Selection Table 3.
Nic Plum	26-Feb-2010	Added consistency rules to SVp-01 for Resource realises Node.
Nic Plum	25-Feb-2010	Added MODAF® trademark. Added minimal modelling process that results from TRAK.
Nic Plum	23-Feb-2010	Improved comparison against MODAF® 1.2 views. Added MODAF 1.2. views not present in TRAK. Added table numbers. Expanded tuples in SVp-05. Fixed duplicate tuples in Optional section for PrVp-03.

Author(s)	Date	Changes
Nic Plum	19-Feb-2010	Original release.

February 2010 Original Release - based on MODAF® 1.2 (and hence also DODAF 1.5).

ACKNOWLEDGEMENTS

This work was originally commissioned by London Underground Ltd.

This Document is based on and incorporates aspects of the Ministry of Defence Architecture Framework MODAF Version 1.2.

A summary of the differences between TRAK Meta-model Version 1 and the MODAF® Version 1.2 can be found at http://trakmetamodel.sourceforge.net. A comparison of the set of TRAK viewpoints[1]/views against the MODAF® 1.2 view set is preserved at the back of this document.

The Document incorporates:

- beta testing and feedback from Joe Silman at the Centre for Railway Research and Education at The of University of Birmingham, UK.
- Human Factors advice and feedback from Christopher Lowe at Liv Systems Ltd.
- advice on viewpoint definition and ISO 42010 from Colin Wood at London Underground Limited
- MODAF® architectural modelling experience, architecture viewpoint definitions & metamodel relationships - Nic Plum at Eclectica Systems Ltd for London Underground Ltd.

[1] TRAK uses 'viewpoint' and 'view' in accordance with ISO/IEC 42010. A MODAF viewpoint is a collection of views.

1 INTRODUCTION / SCOPE

This represents part of logical definition TRAK, an enterprise architecture framework. It provides a means of describing the architecture of systems and is based on the requirements of ISO/IEC/IEEE 42010.

TRAK allows you to describe an enterprise, a concept, a solution (and its procurement) and an architecture task. In ISO/IEC terms each is a 'system of interest' and has stakeholders who have concerns that need to be addressed through the resulting architecture description.

TRAK is solution or implementation free i.e. any UML profile or template is one possible solution to this set of logical requirements in producing TRAK-compliant architecture views and may contain tool or implementation-specific artefacts or constructs. For example the set of attributes that any metamodel element has is important, the inheritance in terms of how this is implemented isn't.

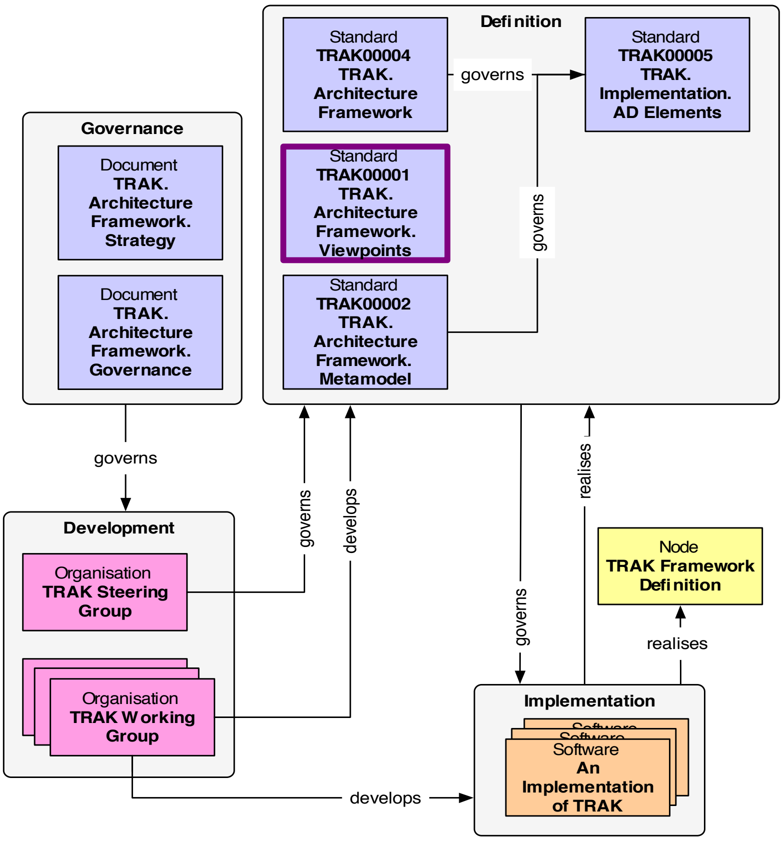

Definition
Standard TRAK00004 TRAK. Architecture Framework
governs
Standard TRAK00005 TRAK. Implementation. AD Elements
Standard TRAK00001 TRAK. Architecture Framework. Viewpoints
governs
Standard TRAK00002 TRAK. Architecture Framework. Metamodel
Governance
Document TRAK. Architecture Framework. Strategy
Document TRAK. Architecture Framework. Governance
governs
governs
develops
realises
Development
Organisation TRAK Steering Group
Organisation TRAK Working Group
governs
Node TRAK Framework Definition
realises
Implementation
Software An Implementation of TRAK
develops

Figure 1-1-Context for the TRAK Architecture Viewpoints Document (This Document)

There are 3 parts to the logical definition of TRAK:-

- TRAK. Architecture Framework. Defines and describes TRAK as a whole and invokes the TRAK Metamodel and TRAK Viewpoints documents. It explains important ideas, provides a common glossary, defines rules that apply to colour and presentation. It also provides guidance on choice of a language to represent TRAK. It defines how TRAK aligns with ISO/IEC/IEEE 42010 and what conformance with TRAK means. It defines a minimum modelling process

- TRAK Metamodel. Defines the metamodel elements and their attributes and the relationships between them. This provides the set of "things" from which a TRAK architecture description is constructed and how they are connected.

- **TRAK Viewpoints - this document**. Defines for each TRAK architecture view, what questions/concerns are addressed by each, what relationships from the TRAK metamodel must and should be used, what is the minimum acceptable content and presentation and what consistency rules apply. This follows the ISO 42010 standard for architecture viewpoints

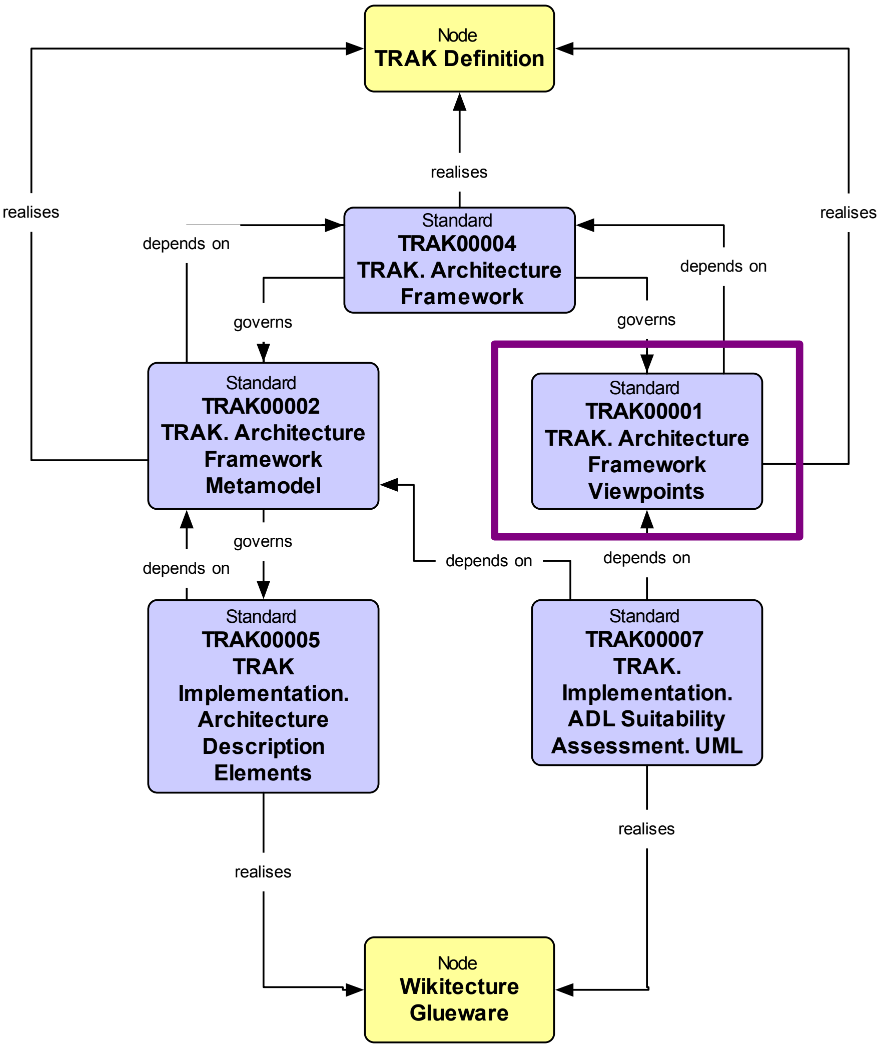


Figure 1-2-Normative TRAK Documents - Logical Definitions vs Implementation of TRAK

At any time there may be many implementations of the logical definition of TRAK. These might realise TRAK in a particular modelling tool or a particular Architecture Description Language (ADL - see glossary). A list of known implementations of TRAK is maintained on Sourceforge, An implementation might implement TRAK in full or only partially. Equally an implementation might introduce its own limitations or artefacts. It is hoped that any implementation will identify any limitations or artefacts that it introduces. If this is done it will help users of TRAK understand what is a product of the TRAK definition and what is a product of the implementation of TRAK using an architecture description language (e.g. UML, BPMN, ArchiMate) or a tool..

All implementations of TRAK shall comply with TRAK. Implementation. Architecture Description Elements. This defines how names of the TRAK metamodel elements and attributes, enumerated lists and applicable standards are to be implemented and is an essential part of assuring consistency of implementation of TRAK. As this is a normative document it is represented as a Standard in TRAK in Figure 1-2.

2 TRAK ARCHITECTURE VIEWPOINTS

Introduction

This section defines each TRAK architecture viewpoint (see glossary) in accordance with ISO/IEC/IEEE 42010 [1]. (hereafter shortened to 'ISO 42010').

The viewpoint stakeholders are selected from the following generic stakeholders within ISO 42010:

- user
- operator
- acquirer
- owner
- supplier
- developer
- builder
- maintainer

with the addition of:

- auditor
- regulator
- trainer
- disposer

In TRAK terms these are roles and a jobholder may perform many roles.

ISO 42010 identifies stakeholders of the system (of interest) which in terms of the TRAK viewpoints are:

- the enterprise
- the concept
- the solution
- the architecture task

and therefore the owner, user, builder of one is not necessarily the same as that of any of the other scopes.

Viewpoint Identification

In TRAK each viewpoint has a '**p**' in the viewpoint identifier to distinguish it from the architecture view that it specifies. For example, the SV**p**-01 specifies the SV-01 Solution Structure Viewpoint. 'A SV-01' is a generic reference to a view type that conforms to the SVp-01 viewpoint.

The viewpoint identifier also defines what architecture perspective the viewpoint (and view) belong to.

- 'EV' - denotes Enterprise Perspective
- 'CV' - denotes Concept Perspective
- 'PrV' - denotes Procurement Perspective
- 'SV' - denotes Solution Perspective
- 'MV' - denotes Management Perspective

There is no sequence for constructing architecture views implied through the viewpoint identifier. There are, however, minimum allowed view sets forming an architecture description depending on which view is constructed owing to the concept of master architecture views.

Titles have been deliberately kept short and, where possible, keyed to the TRAK metamodel element name that dominates the view(point). In other cases a common systems engineering activity is used e.g. mapping interfaces to functional requirements (SVp-03).

Views that show structure have also been separated from those that show exchanges or mappings. All the '-01' views (EV-01, CV-01, PrV-01, SV-01, MV-01) show a form of structure.

Other architecture frameworks such as MODAF® and DODAF may have similarly-named architecture views. Where there is a risk of confusion or a need to disambiguate a TRAK:: namespace must be used. For example the MODAF::OV-4 maps onto the TRAK::SV-01 and TRAK:SV-02 depending on whether organisation structure or resource interaction needs to be shown.

Viewpoint Selection

Viewpoints and therefore architecture views are selected on the basis of the questions or concerns that need to be addressed in accordance with the minimal process from ISO/IEC/IEEE 42010. A list of the 21[2] TRAK viewpoints, each defining a view type, with the concerns each addresses is provided below.

Architecture Perspective	Architecture Viewpoint	Concerns Addressed
Enterprise	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	When is capability required? Is this capability realised by any solutions? Are there any gaps?
Concept	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?
Procurement	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?

[2] Other frameworks such as DODAF, MODAF, NAF and DNDAF have typically twice as many view types.

Architecture Perspective	Architecture Viewpoint	Concerns Addressed
	PrVp-03 Procurement Responsibility	What responsibilities do organisations or jobs have in relation to a project or time? Are their boundaries clear?
Solution	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 operational 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?

Architecture Perspective	Architecture Viewpoint	Concerns Addressed
	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?
Management	MVp-01 Architecture Description Dictionary	Is the architecture description 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 enterprise, concept, procurement, solution or architecture task?
	MVp-04 Assurance	What are the claims made? What is the basis of the claim? Is the claim supported by evidence?

Table 2-1 - TRAK Viewpoints - Concerns Addressed

Anatomy of a Viewpoint

Each TRAK Viewpoint has the following structure:

- Identification number and title. See viewpoint identification.
- Version Number and Date. Each viewpoint has its own version number and date for identification.
- Description. An overview of the viewpoint.
- Concerns Addressed. The stakeholder concerns that the viewpoint addresses. Styled in the form of questions and linked to the typical stakeholders for that viewpoint. Used to enable the right viewpoints to be selected for the architecture task. Note that the stakeholders belong to different subject areas of interest e.g. enterprise, concept, solution and architecture task.
- Anti-Concerns. Concerns that the viewpoint is not suited to addressing.
- Declared Tuples. Tuples (see Glossary in the TRAK Architecture Framework document) from which the responding architecture view must be created. A declared tuple is also an allowed tuple. In effect the resulting architecture view declares these tuples. Derived from TRAK Metamodel.
- Optional Tuples. Tuples that add context or that may appear in any view e.g. Concern *about,* Document *traces to* etc. Only tuples from the mandatory and optional sets may appear on the view. Derived from TRAK Metamodel.
- Well-Formedness. The minimum acceptable view content. Derived from TRAK Metamodel using the mandatory and optional tuple sets. Note that the TRAK Bye Laws for architecture descriptions and views also apply. See overall TRAK document.
- Presentation Methods. Acceptable methods or styles of presenting the view content.
- Views Needed to Construct. Owing to the need to make each element visible within an architecture description (TRAK Bye Laws) and each metamodel element having a master architecture view in which it is 'declared' there is an order in the creation of some TRAK architecture views.
- Consistency Rules. Restrictions on elements and relationships in the view needed to keep it consistent with other views within the architecture description.
- Comments. Additional narrative for guidance.

EVp-01 Enterprise Goal

Version Number

7

Date

8 December 2017

Description

Describes the enterprise and any constituent parts (including by time) together with the goals that the Enterprise Aspires to.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Enterprise Builder, Developer, Maintainer, Owner	What is the Enterprise and what goals does it set out to achieve?

Table 2-2 -EVp-01 Stakeholder Concerns

Anti-Concerns

What is the actual structure / how is the Enterprise realised in practice e.g. directorates ? [i.e. solution perspective]

Declared Tuples

- Enterprise *aspires to* Enterprise Goal
- Enterprise *has part* Enterprise
- Enterprise Goal *has part* Enterprise Goal
- Enterprise Goal *is quantified by* Metric

Optional Tuples

Context

- Organisation *realises* Enterprise

Universal

- Claim *about*, Concern *about, traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

An EV-01 view shall contain:

- at least one Enterprise (the subject of the view)
- every Enterprise must have at least one Enterprise Goal (using Enterprise *aspires to* Enterprise Goal)

If the task stakeholder is concerned with quantifying the enterprise goals (recorded in the MV-02):

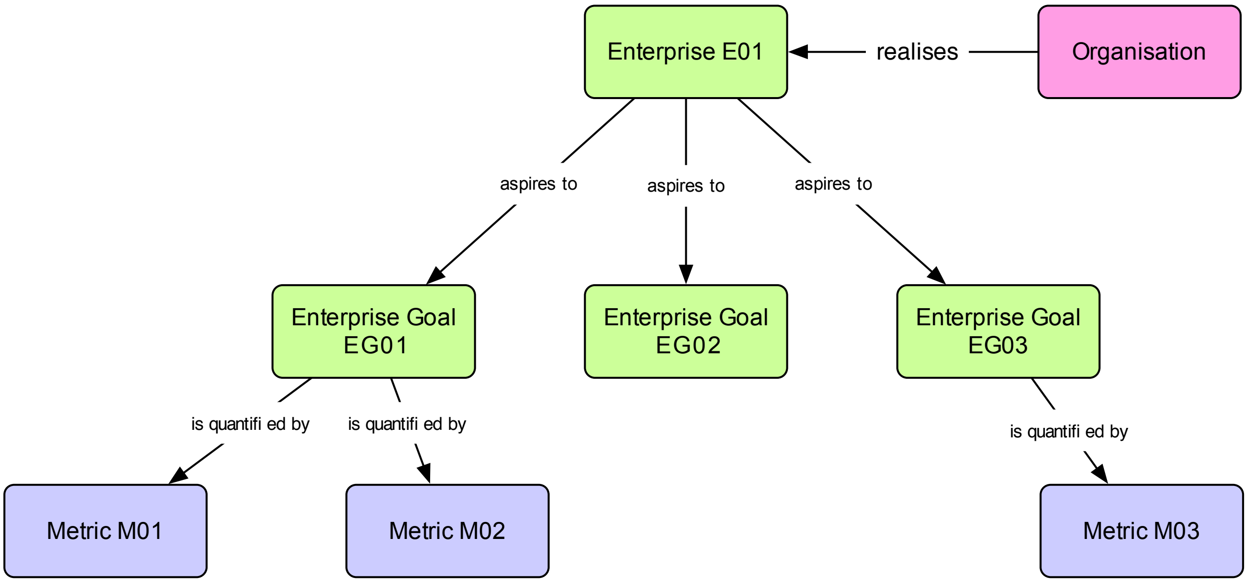
- every Enterprise Goal must have at least one Metric (using Enterprise Goal *is quantified b*y Metric)

Presentation Methods

- block diagram (blocks to represent Enterprise, Enterprise Goal and Metric)

Views Needed In Order to Construct

None - EV-01 is master architecture view for Enterprise and Enterprise Goal.



See minimum TRAK architecture description view sets.

Consistency Rules

Comments

EVp-02 Capability Hierarchy

Version Number

11

Date

8 December 2017

Description

Describes the capabilities required by the Enterprise / Enterprise Goal(s) and dependencies on other Capabilities.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Enterprise Builder, Developer, Maintainer, Owner	What are the enduring capabilities the enterprise requires? How is capability measured?

Table 2-3 EVp-02 Stakeholder Concerns

Anti-Concerns

-

Declared Tuples

- Enterprise *requires* Capability
- Enterprise Goal *requires* Capability
- Capability *depends on* Capability
- Capability *is quantified by* Metric
- Metric *has part* Metric

Optional Tuples

Context

- Organisation *realises* Enterprise
- Enterprise Goal *is quantified by* Metric

Universal

- Claim *about*, Concern *about, traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, traces to Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

An EV-02 view shall contain:

- at least one Enterprise (the subject of the view)
- the subject Enterprise must *require* at least one Capability (using Enterprise *requires* Capability)

If the subject Enterprise has one or more Enterprise Goals:

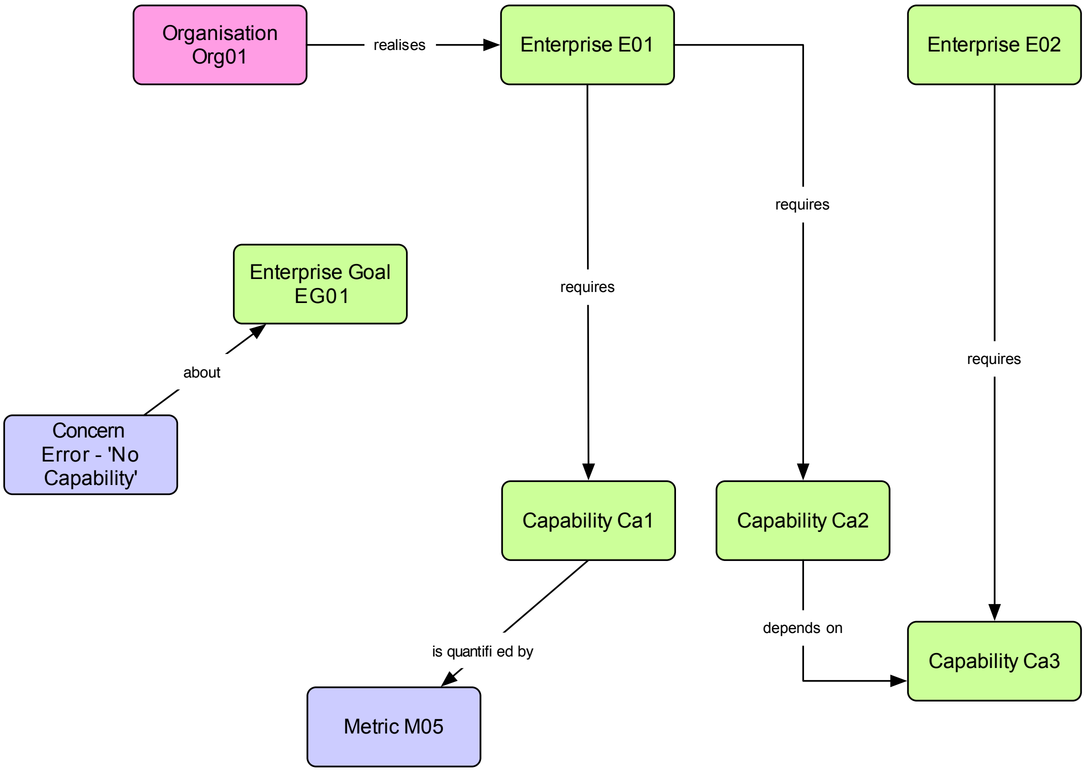
- each Enterprise Goal must *require* at least one Capability (using Enterprise Goal *requires* Capability)

If the task stakeholder is concerned with quantifying capability (recorded in the MV-02):

- at least one Metric (using Capability *is quantified by* Metric)

Presentation Methods

- block diagram (blocks to represent Enterprise, Capability, Metric and Enterprise Goal)



Views Needed In Order to Construct

- EV-01 - master architecture view for Enterprise, Enterprise Goal

See minimum TRAK architecture description view sets.

Consistency Rules

- Enterprise must appear in EV-01

Comments

The creation of taxonomies is more concerned with organisation of collections of any architecture description element, not just capability, and associated with repository management.

Taxonomy diagrams of any type of architecture description element can be included with an architecture description either as a non-conforming product where non-TRAK elements are used (see Conformance with TRAK in TRAK Enterprise Architecture Description document) or using the relevant master architecture view for the TRAK element and the MV-01 definitions to support.

EVp-03 Capability Phasing

Version Number

9

Date

8 December 2017

Description

Describes when Capabilities (planned or actual) are fielded over time. Capabilities are enduring since on their own they have no associated time. An Enterprise has a start and finish date and therefore when a capability is tied to an enterprise this defines a period for which that capability is required. Similarly a system can realise a capability and when delivered or removed by a project activity there is a time period during which the required capability is realised.

The EV-03 view can be used to show the capabilities needed, the capabilities realised (via the solution and procurement perspectives) or contrast the two to determine capability gaps. The EV-03 view also describes dependencies between capabilities.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Enterprise Acquirer, Builder, Developer, Disposer, Maintainer, Owner	When is capability required Is this capability realised by any solutions? Are there any capability gaps?

Table 2-4 EVp-03 Stakeholder Concerns

Anti-Concerns

-

Declared Tuples

[capabilities needed]

- Enterprise *requires* Capability

[capabilities realised]

- Project Activity *delivers* System
- Project Activity *removes* System

- System *realises* Capability

Optional Tuples

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, traces to Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

An EV-03 view shall contain:

[capabilities needed - phasing]

- at least one Enterprise (the subject of the view)
- every Enterprise must have at least one Capability (using Enterprise *requires* Capability)
- the Enterprise must have both a start and a finish date (attributes of TRAK::Enterprise - see TRAK Metamodel document). If these aren't specified it is assumed to be enduring (exists at all dates)
- the time period during which a Capability is required must be visible (as the point is to show change / dependency with time).

[capabilities realised]

In addition if the capabilities realised are to be shown:

- at least one System
- every System must realise at least one of the Capabilities from the [capabilities needed – phasing] above (using System *realises* Capability)
- every System must be associated with 2 Project Activities (one to introduce the System into service, the other to remove it from service) (using Project Activity *delivers* System, Project Activity *removes* System)
- every Project Activity must have both a start and a finish date (attribute of TRAK::Project Activity - see TRAK Metamodel document).

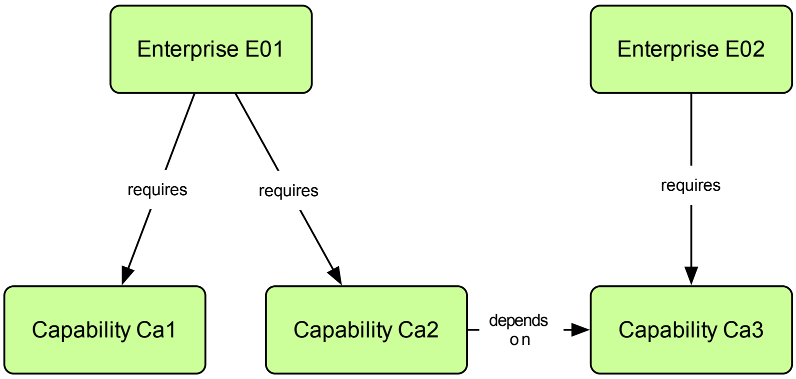
Presentation Methods

- table or matrix ([capabilities needed] Capability vs time taken from start and finish dates for each Enterprise; i.e. Capability vs Enterprise [capabilities realised] Capability vs time taken from start and finish dates for each Project Activity that delivers/removes the Capability - i.e. Capability vs System; or contrast the two). The important feature is to be able to see how capability changes with time.

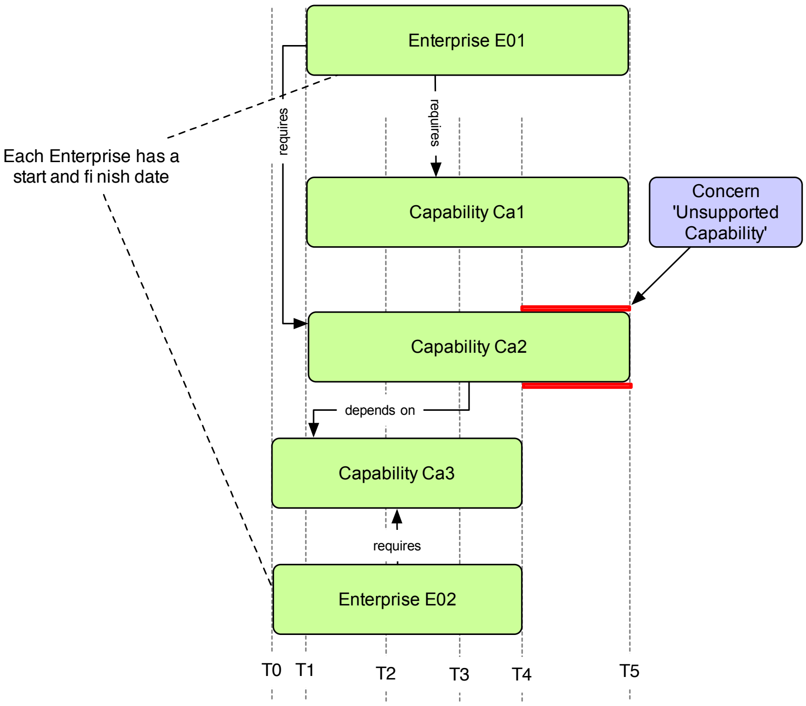
Capability	Required by Enterprise	Start Date	Finish Date	Depends on Capability
Ca1	E01	T1	T5	-
Ca2	E01	T1	T5	Ca3
Ca3	E02	T0	T4	

- block diagram (connected blocks to represent Enterprise, Capability, Project Activity and System):

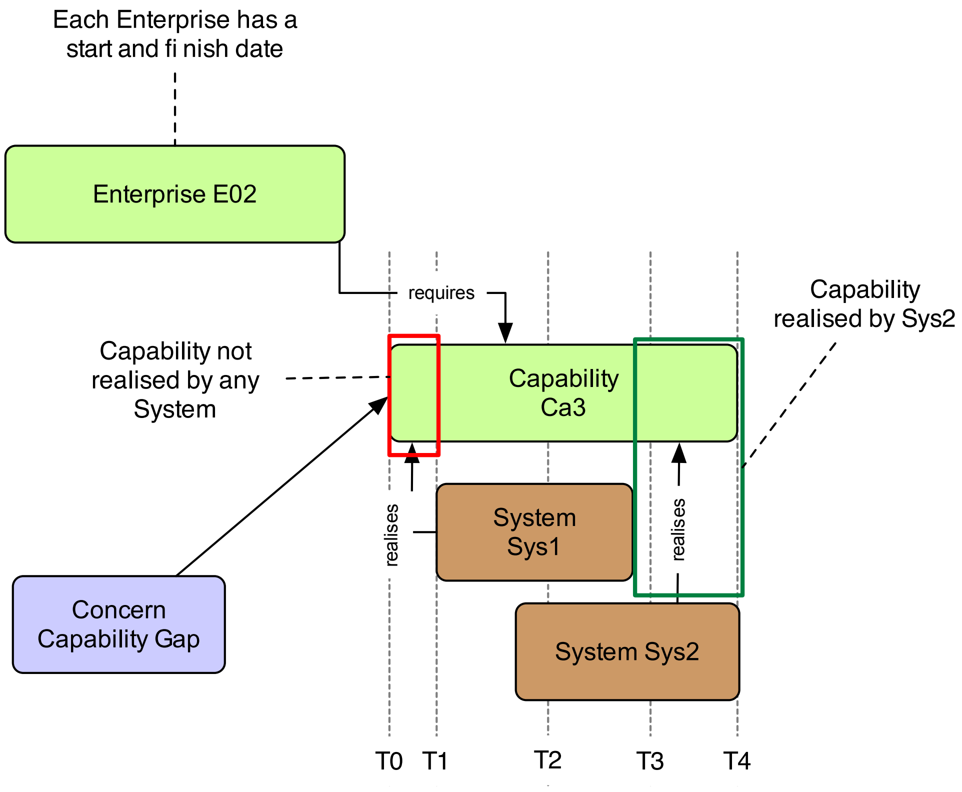
Capabilities needed: - hierarchy (enduring / no time specified)



phasing



capabilities realised


Each Enterprise has a start and fi nish date
Enterprise E02
requires
Capability not realised by any System
Capability Ca3
Capability realised by Sys2
realises
System Sys1
realises
Concern Capability Gap
System Sys2
T0
T1
T2
T3
T4

Views Needed In Order to Construct

- EV-02 - master architecture view for Capability
- PrV-02 - master architecture view for Project Activity
- SV-01 (if System *realises* Capability shown) - master architecture view for System

See minimum TRAK architecture description view sets.

Consistency Rules

- Capability must appear in EV-02
- Project must appear in PrV-01
- System must appear in SV-01

Comments

The time period for the capabilities needed is determined using the start and finish dates for the Enterprise that requires these capabilities.

The time period during which a capability is realised is marked using start and finish date(s) of the Project Activity(ies) which delivers the System(s) that realise the particular capability. If more than one system realises the capability then the period for which the capability is realised is determined by the earliest start date and latest finish date for the project activities involved.

CVp-01 Concept Need

Version Number

11

Date

20 August 2019

Description

Describes the concept in terms of the set of abstract things or "stuff" (nodes) and how they depend on each other by way of conceptual structure and need. The CV-01 provides a way to describe that "A needs B" at a high level that is free from any idea of solution.

The response to a need is an exchange of one or more items which is defined by the CVp-03.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Concept Developer, Disposer, Maintainer, Owner, User	What are the concept needs?

Table 2-5 CVp-01 Stakeholder Concerns

Anti-Concerns

Declared Tuples

- Node *has* Need *for* Node - where Need and Node are blocks, *for* is a connector, or
- Node *needs* Node – where Node is a block, *needs* is a connector
- Node *has part* Node

Optional Tuples

Context - Realisation by Solution

- Resource *realises* Node

where Resource = System, Physical, Software, Organisation, Job or Role

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

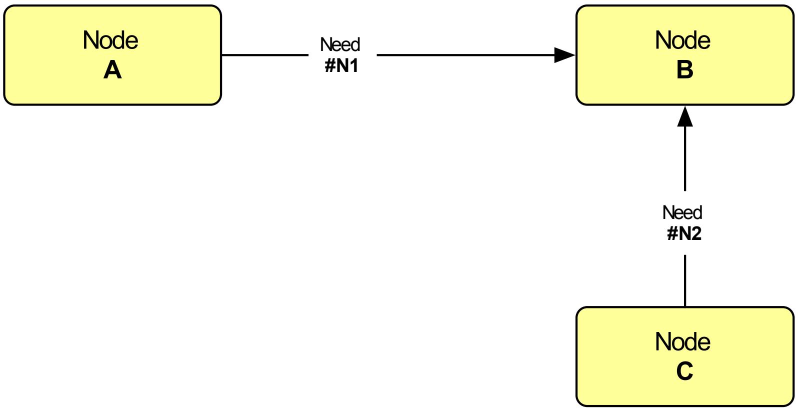
Well-Formedness

A CV-01 view shall contain:

- at least two Nodes (including the subject of the view)
- every Node must be connected to at least one Need (using Node *needs* Node / Node *has* Need *for* Node depending on whether Need is represented as a node or a connector))
- each Need (line or intersection in a matrix) must be uniquely identified, have a direction and a description

Presentation Methods

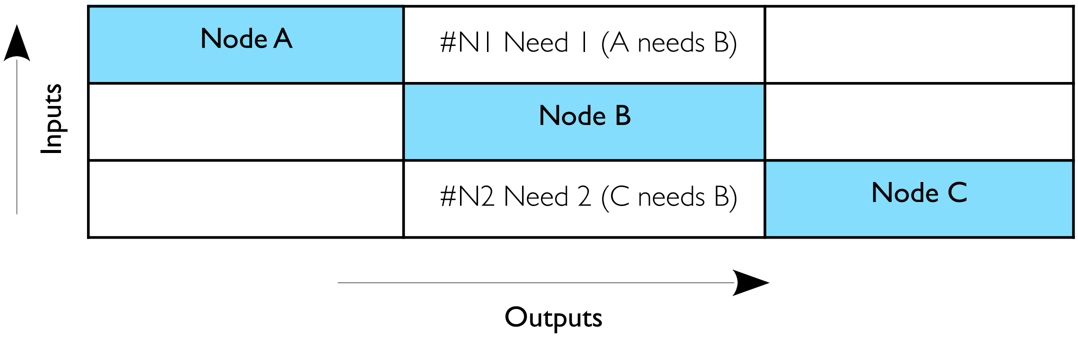
- block diagram (Node = block, Need = line with text label and direction indicator)



- matrix (if only showing need relationships)

Need Identifier	Source Node	Needed Node	Need(s)	Description
#N1	A	B	Need 1	nnn mmm
#N2	C	B	Need 2	mmm sss jj

- N-squared diagram (Nodes on diagonal, intersections marked to represent direction of Need)



Node A	#N1 Need 1 (A needs B)	
	Node B	
	#N2 Need 2 (C needs B)	Node C

Views Needed In Order to Construct

None - CV-01 is the master architecture view for Node, Need

See minimum TRAK architecture description view sets.

Consistency Rules

Comments

The CV-01 is the Master Architecture View for Node, Need.

A Node (see TRAK Metamodel document) is a solution-free "thing" and provides a way of very coarsely describing a concept. Although solution-free if there are real world restrictions or "givens" that the concept has to work with these may be represented as a single thing. No detail or technology!

CVp-03 Concept Item Exchange

Version Number

8

Date

8 December 2017

Description

Describes what is exchanged to meet the solution-free needs identified. It responds to the needs identified in the CVp-01.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Concept Developer, Disposer, Maintainer, Operator, Owner, Trainer, User	What is required to satisfy the concept needs? Have the items exchanged by concept nodes been identified?

Table 2-6 CVp-03 Stakeholder Concerns

Anti-Concerns

-

Declared Tuples

To establish the existence of the exchange:

EITHER

{

- Item Exchange *from* / *to* Node

}

OR

{

- Node *has* Need
- Need *requires* Item Exchange

}

To characterise the exchange:

- Item Exchange *carries* Item
- Item *has part* Item

Optional Tuples

Context - Concept Structure

- Node *has* Need *for* Node (Node *needs* Node)
- Node *has part* Node

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, Standard *governs*, *traces to* Requirement may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

Pre-requisites:

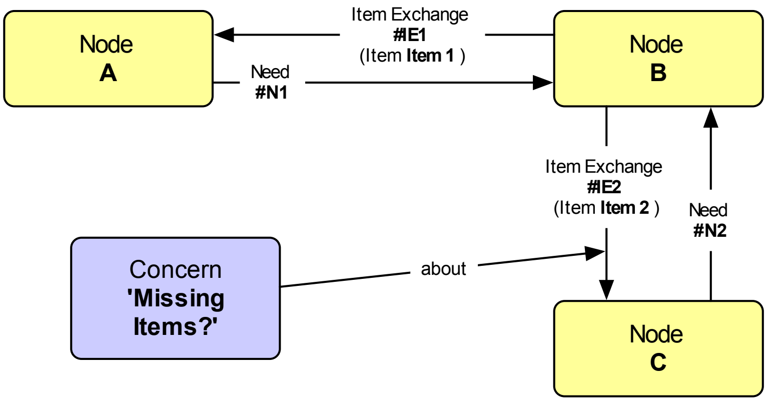
- at least 1 Node-pair (2 Nodes related to each other by 'Node *has* Need *for* Node' or Node *needs* Node) (the subject of the view) must exist on a CV-01

A CV-03 view shall contain:

- at least 1 Node-pair (the subject of the view – using 'Node *has* Need *for* Node' or Node *needs* Node)
- every Node-pair must have at least 1 Item Exchange (using Item Exchange *from* / *to* Node)
- every Item Exchange must be characterised by at least 1 Item being exchanged (using Item Exchange *carries* Item))
- every Item Exchange and Item must be typed (allowed values = Unknown, Data, Energy or Resource - see TRAK Metamodel document). Default= 'Unknown' - a possible concern in the architecture description.
- every Item Exchange must be uniquely identified, have a direction and a description
- every Item Exchange between a node pair has a direction (from / to) which is opposite to the direction of the Need for the same node pair
- every Node must be uniquely identified

Presentation Methods

- block diagram (Node = block, Item Exchange = line with direction indicator, Item = text label,



Need = line with direction indicator)

Note: The item exchanges are typed = 'IE1=Energy, IE2=Data'. If this cannot be attached to the resource interaction(s) an accompanying list of typed item exchanges would be needed.

- Matrix

Item Exchange Identifier	Source Node	Destination Node	Item(s)	Requiring Need	Description
#IE1	B	A	Item 1 (Energy)	#N1	cc bsms
#IE2	B	C	Item 2 (Data)	#N2	nsn,m oe

- N-squared diagram (Nodes on diagonal, intersections marked with Item(s) to represent direction of Item Exchanges and Items being exchanged)

Inputs

Node A		
#IE1 Item 1 (from B to A) (Energy)	Node B	#IE2 Item 2 (from B to C) (Data)
		Node

Outputs

Views Needed In Order to Construct

- CV-01 - the master architecture view for Node, Need.

See minimum TRAK architecture description view sets.

Consistency Rules

- Node must appear in CV-01

Comments

The CV-03 is the Master Architecture View for Item, Item Exchange.

Note that the direction in which the item exchange occurs is in the opposite direction to the need described in CVp-01 because the exchanges are only required because they satisfy the need(s) previously identified.There is therefore a mapping between a need and 1 or more item exchanges.

CVp-04 Concept Activity to Capability Mapping

Version Number

7

Date

8 December 2017

Description

Describes how the concept activities relate to the enterprise capabilities needed. This enables the following problems to be identified:

- capabilities required by an enterprise that aren't supported by any concept activity.This might lead to nothing being developed to deliver the capability.
- concept activities that aren't needed.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Concept Developer, Owner **Enterprise** Builder, Developer, Maintainer, Owner	How / are concept activities sufficient to deliver capability?

Table 2-7 CVp-04 Stakeholder Concerns

Anti-Concerns

-

Declared Tuples

- Node *conducts* Concept Activity
- Concept Activity *supports* Capability
- Enterprise *requires* Capability

Optional Tuples

- Enterprise Goal *requires* Capability

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

A CV-04 view shall contain:

- at least one Node (the subject of the view)
- every Node must be connected to at least one Concept Activity (using Node *conducts* Concept Activity)
- at least one Enterprise
- every Enterprise must be connected to at least one Capability (using Enterprise *requires* Capability)
- every Concept Activity must be connected to at least one Capability (using Concept Activity *supports* Capability), if not then a Concern *
- every Capability must be connected to at least one Concept Activity (using Concept Activity *supports* Capability), if not then a Concern *

* Note: The purpose of the viewpoint is to identify unmapped capabilities and unmatched concept activities. Unmapped items are to be flagged by connecting to a Concern (using 'Concern *about* Concept Activity', 'Concern *about* Capability.') identifying this concern and reported on in the MV-02. As items are mapped on this view they are disconnected from the Concern.

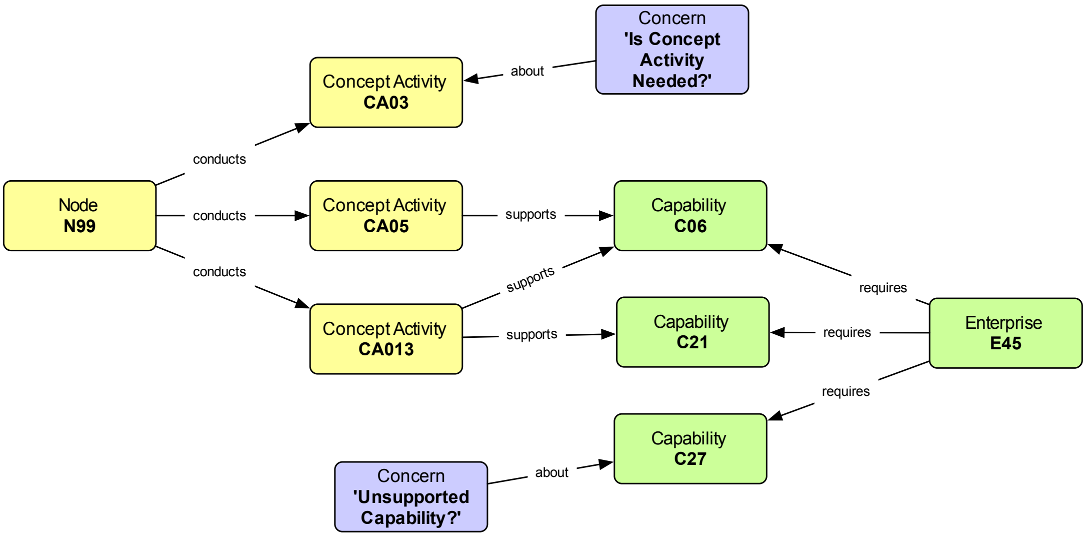
Presentation Methods

- Table or Matrix (row/column headings = Node, Concept Activity, Capability, Enterprise, Description)

Node	Concept Activity	Capability	Enterprise	Description
#N99	CA03	(missing)	(missing)	Concern - is this concept activity needed?
#N99	CA05	C06	E45	mmm sss jj
#N99	CA013	C06	E45	mmm sss jj

Node	Concept Activity	Capability	Enterprise	Description
#N99	CA05	C21	E45	mmm sss jj
(missing)	(missing)	C27	E45	Concern - no concept activity identified to support this capability.

- Block diagram (Node, Concept Activity, Enterprise, Capability = block, TRAK relationship = line with direction indicator)



Note: Any unmapped concept activity or capability as far as the subject pair (particular Node and Enterprise) being described represents a concern.

Views Needed In Order to Construct

- CV-01 - master architecture view for Node
- CV-05 - master architecture view for Concept Activity
- EV-02 - master architecture view for Capability
- EV-01 - master architecture view for Enterprise

See minimum TRAK architecture description view sets.

Consistency Rules

- Node must appear in CV-01
- Concept Activity must appear in CV-05

- any Standard must appear on MV-03

Comments

As a mapping view, the CV-04 is not a master architecture view.

CVp-05 Concept Activity

Version Number

10

Date

20 August 2019

Description

Describes what the node ("thing") does. This provides a way of describing the overall concept in functional terms that are free of solution - i.e what it needs to do without the "how".

Concerns Addressed

Stakeholder	Concern of Stakeholder
Concept Developer, Disposer, Maintainer, Operator, Owner, Trainer, User	What does each node need to do?

Table 2-8 CVp-05 Stakeholder Concerns

Anti-Concerns

-

Declared Tuples

- Node *conducts* Concept Activity
- Concept Activity *has part* Concept Activity
- Concept Activity *is quantified by* Metric
- Metric *has part* Metric

Optional Tuples

- ~~Item *has part* Item~~
- Function *realises* Concept Activity
- Concept Activity *supports* Capability
- Node *has part* Node

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

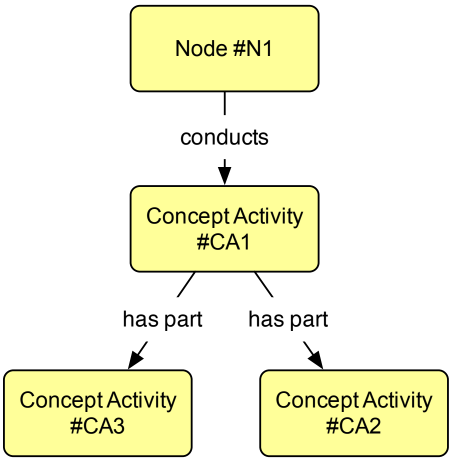
A CV-05 view shall contain:

- at least one Node (the subject of the view)
- the subject Node must have at least one Concept Activity (using Node *conducts* Concept Activity)

Presentation Methods

- block diagram (Node, Concept Activity, Item = block, TRAK relationship = line with direction indicator)

(hierarchical form)



Views Needed In Order to Construct

- CV-01 - master architecture view for Node

See minimum TRAK architecture description view sets.

Consistency Rules

- Node must appear in CV-01

- **IF** {Function *realises* Concept Activity} **THEN** {Resource (that *performs* same Function) *realises* Node must be made on the SV-01 }

Comments

The CV-05 is the master architecture view for Concept Activity.

If ordering of Concept Activities is a concern use a Concept Sequence View defined by the CVp-06.

CVp-06 Concept Sequence

Version Number

8

Date

8 December 2017

Description

Describes the order in which logical events or concept activities occur.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Concept Developer, Disposer, Maintainer, Operator, Owner, Trainer, User	How are concept activities ordered? Is it important?

Table 2-9 CVp-06 Stakeholder Concerns

Anti-Concerns

-

Declared Tuples

Functional Sequence

- Node *conducts* Concept Activity
- Concept Activity *triggers* Item
- Item *triggers* Concept Activity
- Concept Activity *precedes* Concept Activity

Exchange Sequence

- Item Exchange *from* / *to* Node
- Item Exchange *carries* Item

Optional Tuples

Context

- Item *has part* Item
- Concept Activity *has part* Concept Activity

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

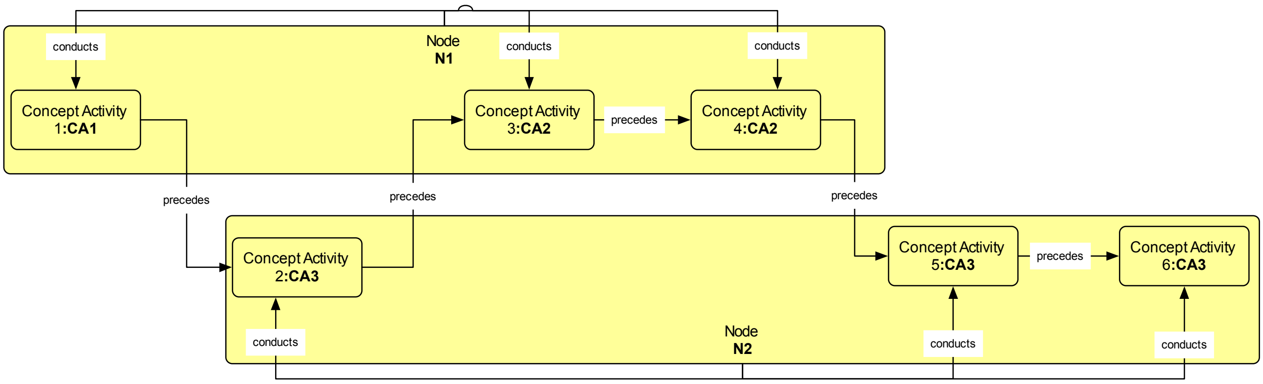
A CV-06 view shall contain:

- at least one Node (the subject of the view)
- the subject Node must have at least 2 Concept Activities (using Node *conducts* Concept Activity)
- every Concept Activity must be connected to another Concept Activity **EITHER** by (Concept Activity *triggers* Item triggers Concept Activity) **OR** (Concept Activity *precedes* Concept Activity)
- a means of identifying the order in which all Concept Activities occur e.g. by numbering or explicit layout on a time line

Presentation Methods

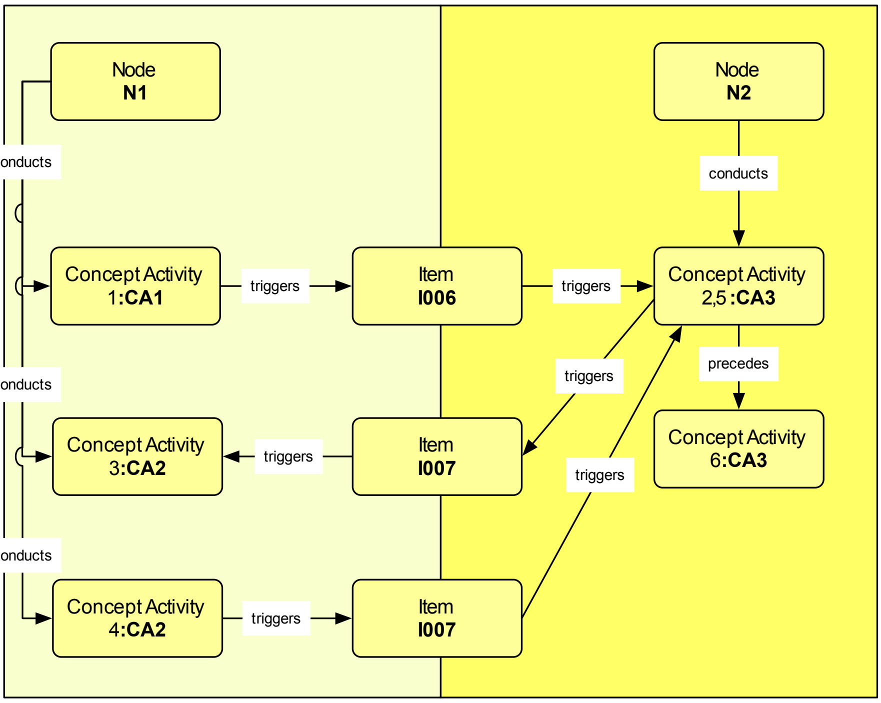
- sequence diagram
- activity diagram
- collaboration diagram

Functional Sequence:



Only describes/needs concept activity order.

Item Sequence



Describes ordering in context of item exchanges.

Views Needed In Order to Construct

- CV-01 - master architecture view for Node

See minimum TRAK architecture description view sets.

Consistency Rules

- any Node must appear on a CV-01

- any Concept Activity must appear on a CV-05
- any Items involved in triggers on the CV-06 must be consistent with those defined in Item Exchanges on the CV-03 i.e.
  - **IF** (Node *conducts* Concept Activity *triggers* Item on CV-06) **THEN** (Item Exchange *from* (same) Node *carries* (same) Item on CV-03)
  - **IF** (Item *triggers* Concept Activity *conducted by* Node CV-06) **THEN** (Item Exchange *to* (same) Node *carries* (same) Item on CV-03)

Comments

PrVp-01 Procurement Structure

Version Number

7

Date

8 December 2017

Description

Describes the structural organisation for procurement of the solution(s).

Concerns Addressed

Stakeholder	Concern of Stakeholder
Enterprise Builder, Owner **Solution** Acquirer	What is the project structure? How is it governed?

Table 2-10 PrVp-01 Stakeholder Concerns

Anti - Concerns

-

Declared Tuples

- Project *has part* Project
- Organisation *governs* Project

Optional Tuples

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

A PrV-01 view shall contain:

- at least one Project (the subject of the view)
- any project parts are connected together using Project *has part* Project

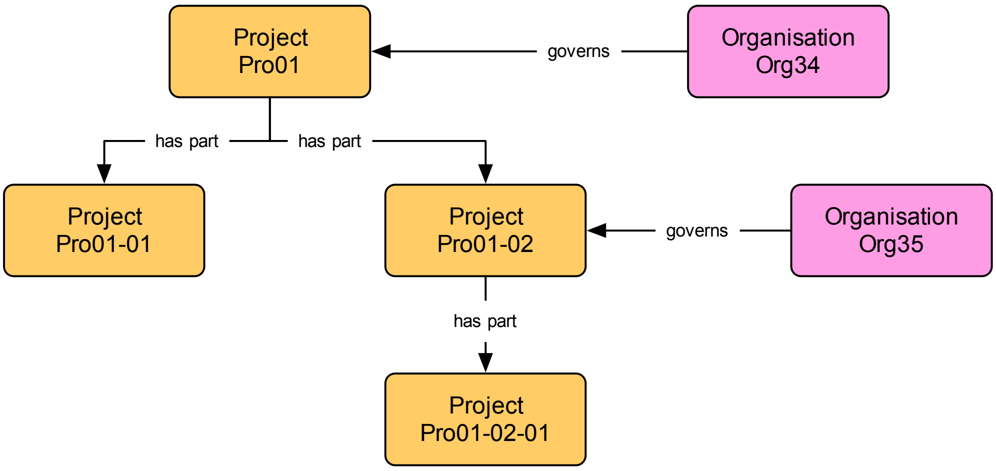
If it is known what organisation (or part of an organisation) is in control of the project then:

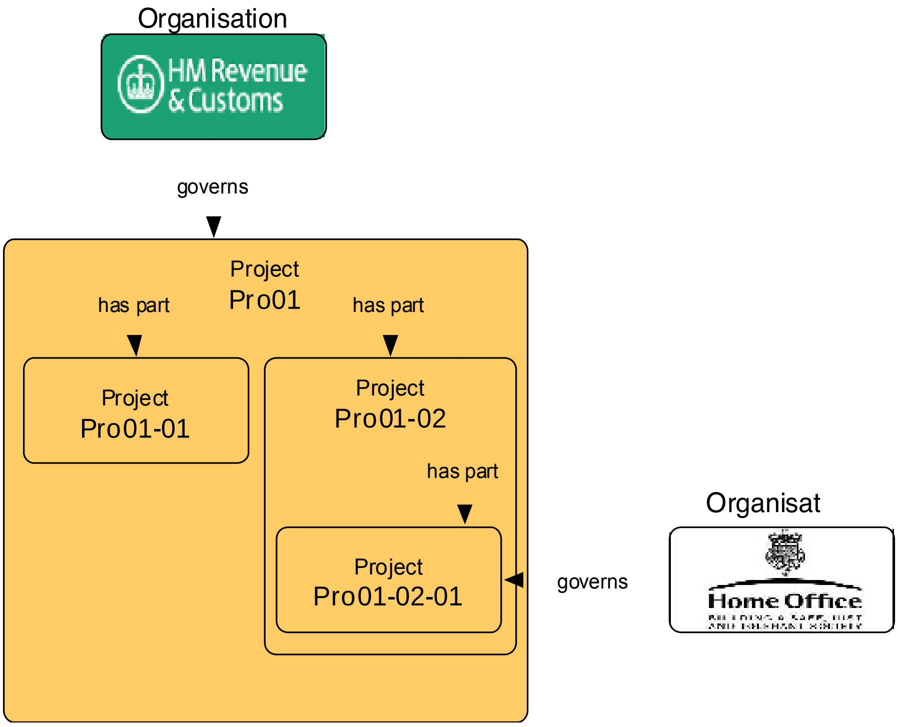
- every Project must be connected to an Organisation using Organisation *governs* Project

Note: Where there is a project hierarchy structure it is assumed that the same organisation governs every project in the hierarchy below the point at which the relationship is made unless an additional explicit relationship is shown. This removes the need to connect every project in the hierarchy.

Presentation Methods

- block diagram (Project, Organisation = block, TRAK relationship = line with direction indicator)





Views Needed In Order to Construct

- SV-01 (if Organisation governs Project) - master architecture view for Resource

See minimum TRAK architecture description view sets.

Consistency Rules

If 'Organisation *owns* Project' is made in the PrV-01:The Organisation that governs the Project cannot itself be part of the Solution since the former is part of the model of the business and the latter will be part of the model of the thing delivered or developed. The governing organisation might well provide the resource to deliver the project but it is a different organisation to that in the solution - it is important to keep these 2 models distinct.

{ organisation in Organisation *governs* Project}

NOT EQUAL

- { organisation in

{ EITHER

- (same) Project *undertakes* Project Activity *delivers* / *removes* (System *is configured with* Organisation)

OR

- (System *is configured with* Organisation) *is necessary for* Project Activity **AND** (same) Project *undertakes* Project Activity

OR

- (same) Project *owns* Milestone *marks introduction of* / *marks removal of* (System *is configured with* Organisation)

}

}

This also comes into play if System *is configured with* Organisation is made in the SV-01,

Comments

It does for Project what the SV-01 does for Resource in the solution perspective.

The PrV-01 is the master architecture view for Project.

PrVp-02 Procurement Timeline

Version Number

8

Date

8 December 2017

Description

Describes the dependencies between projects as a result of the introduction or removal of systems

Concerns Addressed

Stakeholder	Concern of Stakeholder
Enterprise Builder, Owner **Solution** Acquirer, Developer, Disposer, Maintainer, Operator, Owner, Trainer	What other projects is this dependent on? How does their delivery time affect us?

Table 2-11 PrVp-02 Stakeholder Concerns

Anti - Concerns

-

Declared Tuples

To establish the relationship between Project and System:

- Project *owns* Milestone
- Project Activity *marked by* Milestone
- Milestone *marks introduction of* System
- Milestone *marks removal of* System

OR

- Project *undertakes* Project Activity
- Project Activity *delivers* System

- Project Activity *removes* System

OR

- Project *undertakes* Project Activity
- System *is necessary for* Project Activity

Optional Tuples

Context - Project Structure

- Project *has part* Project

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

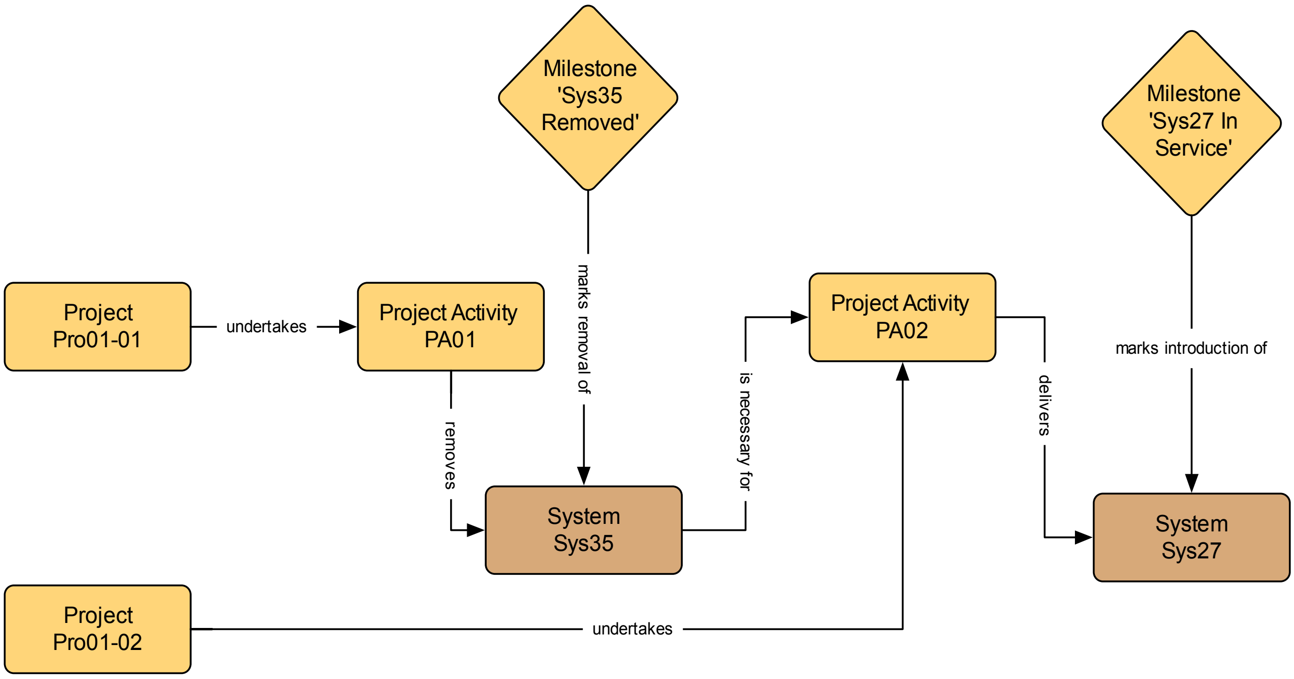
Well-Formedness

A PrV-02 view shall contain:

- at least 2 Systems OR at least 2 Project Activities (trying to show dependencies between project activities via the system(s)) (using Project Activity *delivers* / *removes* System *is necessary for* Project Activity)
- each System and Project Activity must be connected to its Project (using Project *undertakes* Project Activity *delivers* / *removes* System or Project *owns* Milestone *marks introduction of* / *marks removal of* System)
- at least one System (the subject of the view - it shows the procurement / effect of time on the solution)
- every Project Activity must have start and finish dates or a Concern attached flagging the absence i.e. Concern *about* Project Activity.

Presentation Methods

- gantt chart
- block diagram (Project, Project Activity, Milestone, System = block, TRAK relationship = line with direction indicator)



Note: Since start and finish dates are not shown/accessible only the dependency of Project Pro01-02 on Project Pro01-01 via System Sys35 is known.

Views Needed In Order to Construct

- PrV-01 - master architecture view for Project
- SV-01 - master architecture view for Resource (System)

See minimum TRAK architecture description view sets.

Consistency Rules

- System must appear on SV-01
- Project must appear on PrV-01

The relationships between Milestone and System and Project Activity and System must be consistent:

- **IF** {Project *undertakes* Project Activity *delivers* System} **THEN NOT** {(same) Project *owns* Milestone *marks removal of* (same) System}
- **IF** {Project *undertakes* Project Activity *removes* System} **THEN NOT** {(same) Project *owns* Milestone *marks introduction of* (same) System}
- **IF** {Project *owns* Milestone *marks removal of System*} **THEN NOT** {(same) Project *undertakes* Project Activity *delivers* (same) System}
- **IF** {Project *owns* Milestone *marks introduction of System*} **THEN NOT** {(same) Project *undertakes* Project Activity *removes* (same) System}

Comments

The PrV-02 is the master architecture view for Milestone, Project Activity.

Attributes/properties for Project Activity e.g. start and finish dates are specified in the TRAK Metamodel document.

PrVp-03 Procurement Responsibility

Version Number

7

Date

8 December 2017

Description

Describes the extent of a role at a point in time.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Enterprise Builder, Owner **Solution** Acquirer, Developer, Disposer, Maintainer, Owner, Trainer	What responsibilities do organisations or jobs have in relation to a project or time? Are their boundaries clear?

Table 2-12 PrVp-03 Stakeholder Concerns

Anti - Concerns

-

Declared Tuples

To establish the relationship between Project and System:

EITHER

{

- Project *owns* Milestone
- Project Activity *marked by* Milestone
- Milestone *marks introduction of* System
- Milestone *marks removal of* System

}

OR

{

- Project *undertakes* Project Activity
- Project Activity *delivers* System
- Project Activity *removes* System

}

OR

{

- Project *undertakes* Project Activity
- System *necessary for* Project Activity

}

AND

To establish the responsibility extent via System:

{

← {

- Organisation *plays* Role

← OR

- Job *plays* Role

}

AND

{

- Role *extends to* System
- System *is configured with* Resource (Job, Organisation, Physical, Role, Software, System)

← }

}

Note that as the Project Perspective shows the introduction or removal of System at a time specified by the Project Activity the duration of the Role extent is that of the life of the System (between its introduction and removal).

Optional Tuples

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

A PrV-03 view shall contain:

[to establish the time point/duration]

- one Milestone or one Project Activity (sets the time at which the responsibility applies)
- every Milestone or Project Activity must be connected to 1 Project (using 'Project *owns* Milestone' or 'Project *undertakes* Project Activity' respectively)

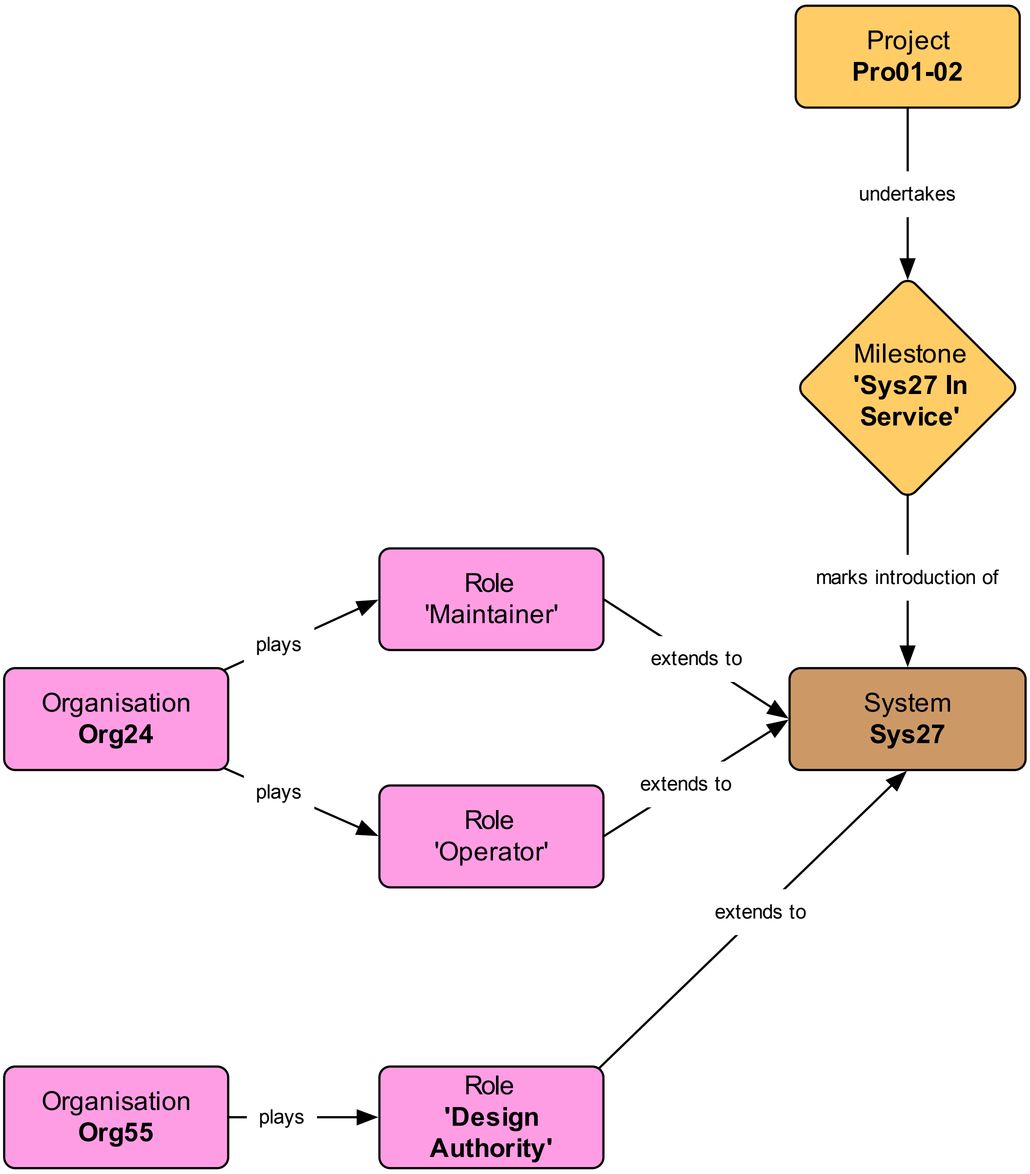
Note: A PrV-02 is needed first to make the relationships between Project, Project Activity/Milestone and System to set 'project time'

[responsibility extent]

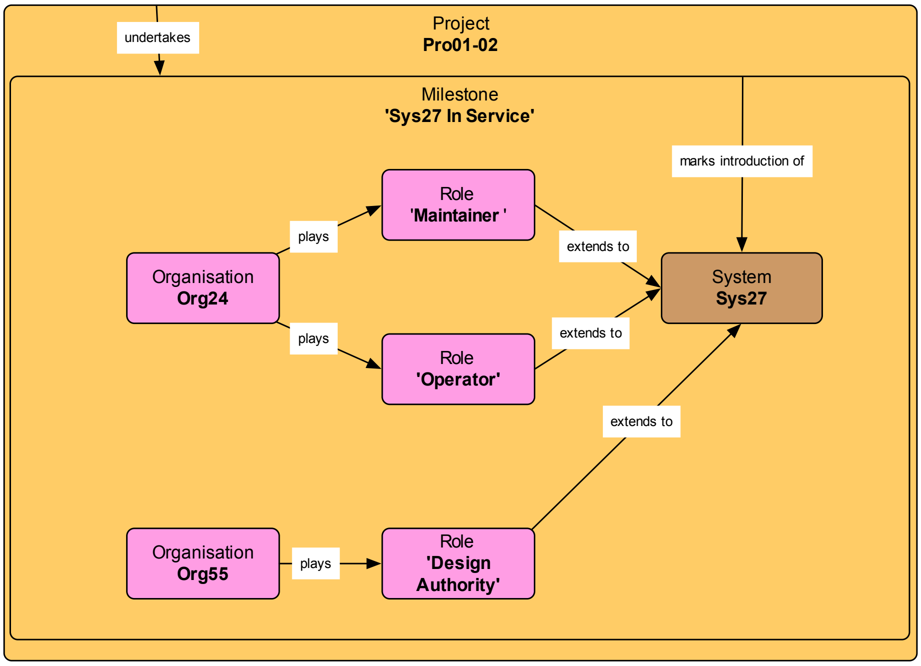
- at least 1 Organisation or at least one 1 Job
- every Job / Organisation must be connected to at least 1 Role (using Job / Organisation *plays* Role)
- every Role must be connected to at least 1 System (using Role *extends to* System)

Presentation Methods

- block diagram (Project, Project Activity, Milestone, Resource = block, TRAK relationship = line with direction indicator)



Nested form:



The nested form is useful because it immediately makes clear that the responsibilities apply at/during a time period formed by the surrounding frame. It is essential that the relationships must be visible (see TRAK Bye Laws in TRAK Enterprise Architecture Framework document).

Views Needed In Order to Construct

- SV-01 - master architecture view for Resource (Job, Organisation, Role, System)
- PrV-01 - master architecture view for Project
- PrV-02 - master architecture view for Milestone, Project Activity

See minimum TRAK architecture description view sets.

Consistency Rules

- Project must appear on PrV-01
- Project Milestone must appear on PrV-02
- any Resource must appear on SV-01
- **IF** System *is configured with* Role **THEN NOT {**(same) Role *extends to* (same) System**}** i.e. the Role that forms the responsibility extent is part of a different system

Comments

If you need to show responsibility extent independently of time use the SV-01 with the Organisation or Job *plays* Role *extends to* Resource construct.

SVp-01 Solution Structure

Version Number

16

Date

31st January 2018

Description

Describes solution structure in terms of parts, governance, membership, dependency and responsibility extent.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Solution Acquirer, Developer, Maintainer, Operator, Owner, Trainer, User	How does responsibility (scope/jurisdiction) apply to the solution components? What does the solution consist of? Is it structured sensibly? What is the organisation structure / membership?

Table 2-13 SVp-01 Stakeholder Concerns

Anti - Concerns

-

Declared Tuples

Configuration

- Software *has part* Software
- Software *hosted on* Physical
- Physical *contains* System
- Physical *physically depends on* Physical
- Physical *has part* Physical
- Physical *contains* Physical
- Physical *is attached to* Physical
- Physical *physically supports* Physical
- Organisation *has part* Organisation

- Organisation *has part* Job
- System *is configured with* Resource

where Resource = System, Physical, Software, Organisation, Job or Role

Governance

- Organisation *is member of* Organisation
- Organisation *governs* Organisation
- Job *governs* Job

Role Extent

- Organisation *plays* Role
- Job *plays* Role
- Role *extends to* Resource

where Resource = System, Physical, Software, Organisation, Job or Role

Solution Realises...

- Resource *realises* Node

where Resource = System, Physical, Software, Organisation, Job or Role

- Organisation *realises* Enterprise
- System *realises* Capability

Optional Tuples

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

A SV-01 view shall contain:

EITHER

[configuration]

- at least the Resource (the subject of the view) connected to another Resource (using System *is configured with Resource*, *Physical has part Physical, Organisation has part Organisation, Software has part Software, Software hosted on Physical, Physical is attached to* / *physically depends on* / *physically supports* / *contains* Physical as permitted by the metamodel)

OR [governance]

- at least an Organisation or a Job (the subject of the view)
- that Organisation or Job connected to another Organisation or Job (using Organisation *governs* / *is member of* Organisation, Job *governs* Job or Organisation *has part* Job)

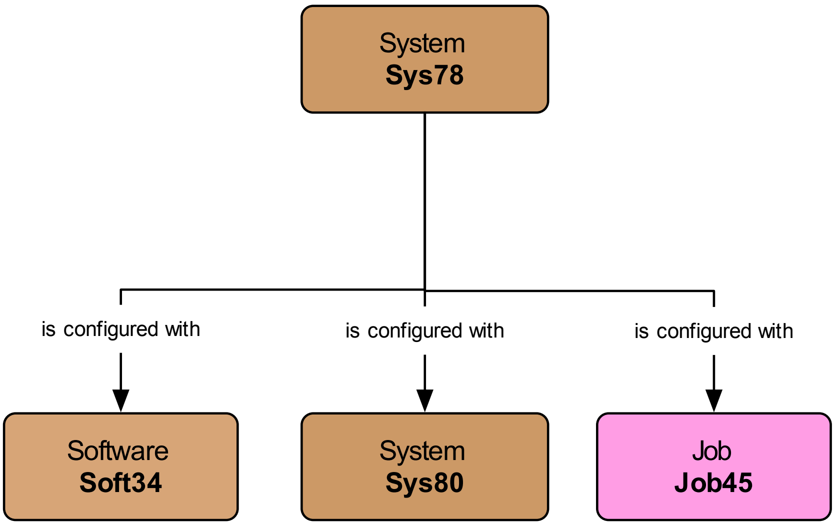
OR [role extent]

- at least 1 Organisation or Job (the subject of the view)
- the subject Organisation or Job connected to at least 1 Role (the subject role) (using Organisation or Job *plays* Role)
- the subject role connected to at least 1 Resource (using Role *extends to* Resource)

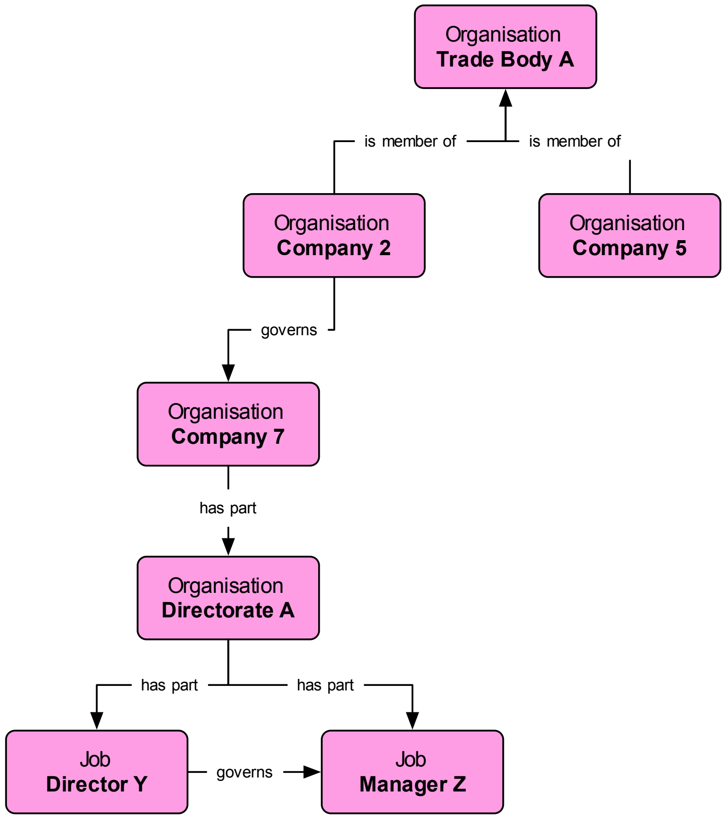
Presentation Methods

- block diagram (Resource = block, TRAK metamodel relationship = line with text label and direction indicator)

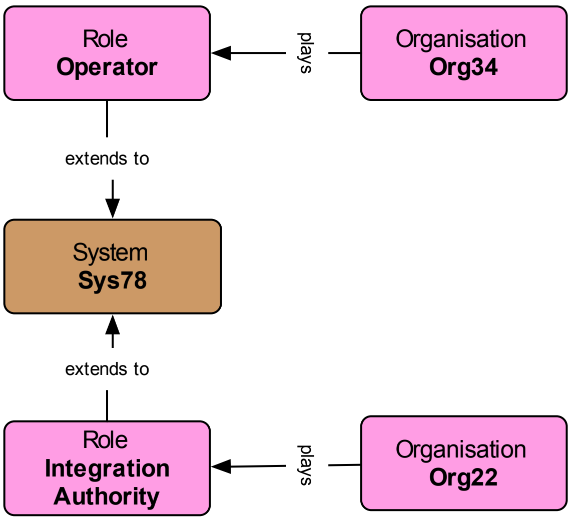
[configuration]



[governance]



[role extent]



Views Needed In Order to Construct

- CV-01 (if System *realises* Node shown) - master architecture view for Node

See minimum TRAK architecture description view sets.

Consistency Rules

- if the SV-01 is used to define the extents of system authority, manufacturer, design authority or independent safety authority using Resource *plays* Role *extends to* (different) Resource these must be consistent with the values of role-based attributes for System (system authority, design authority, manufacturer, independent safety authority) or Resource (design authority, manufacturer, independent safety authority).
- Functional Realisation of Node.: **IF {** Node *conducts* Concept Activity (CV-05) **} AND {** Resource *performs* Function *realises* (same) Concept Activity (SV-04 + SV-05) **} THEN {** Resource *realises* Node **}** must be made on the SV-01.
- Topological Realisation of Node : **IF {** Node *has* Need (CV-01) **} AND {** Resource Interaction *realises* (same) Need (SV-02) **} THEN {** Resource *realises* Node **}** must be made on the SV-01
- If System *is configured with* Organisation is made in the SV-01: The Organisation that governs the Project cannot itself be part of the Solution since the former is part of the model of the business and the latter will be part of the model of the thing delivered or developed. The governing organisation might well provide the resource to deliver the project but it is a different organisation to that in the solution - it is important to keep these 2 models distinct.

{ organisation in Organisation *governs* Project}

NOT EQUAL

{ organisation in

{ EITHER

- (same) Project *undertakes* Project Activity *delivers* / *removes* (System *is configured with* Organisation)

OR

- (System is configured with Organisation) *is necessary for* Project Activity **AND** (same) Project *undertakes* Project Activity

OR

- (same) Project *owns* Milestone *marks introduction of* / *marks removal of* (System i*s configured with* Organisation)

}

This also comes into play if Organisation governs Project is made in the PrV-01.

Comments

The SV-01 is the master architecture view for Resource (Job, Organisation, Physical, Role, Software, System)

The SV-01 provides 3 areas of coverage:

1. Describing organisational structure - governance, membership and roles as well as organisational breakdown.
2. Describing the extent or jurisdiction of roles.
3. Describing classic system or product breakdown structure.

If you need to show the effect of time on responsibility extent use the PrV-03 Procurement Responsibility View

Software *is hosted on* Physical is an implicit statement of containment. To make an explicit statement it is acceptable to also state that Physical *contains* Software.

contains and *has part* are transitive relationships / assertions i.e Physical *contains* System means that any element of the System is also contained by the Physical.

SVp-02 Solution Resource Interaction

Version Number

11

Date

8 December 2017

Description

Describes and characterises resource interactions (energy, materiel, data) between solution resources.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Solution Acquirer, Developer, Disposer, Maintainer, Operator, Owner, Trainer, User	How are resources connected together? How are the organisations, jobs and roles connected? Have the interactions / interfaces / exchanges been characterised? What causes events?

Table 2-14 SVp-02 Stakeholder Concerns

Anti - Concerns

-

Declared Tuples

Always at least Interface Identification.

Note that not all theoretical combinations are likely or interesting from a systems engineering point of view.

Interface Identification

Identifying the Resource Interaction between a pair of Resources requires 1 of the following tuples:

{

- Resource – *from* / *to* (Resource Interaction) - Resource

where Resource = System, Physical, Software, Organisation, Job or Role

}

and may extend this to include, if known:

- Resource Interaction carries Interaction Element

Interface Characterisation

The other form builds on this to characterise the Resource Interaction between a pair of Resources:

Interface Identification+

At least a 2 of the following tuples to add Ports to the Resources to be characterised:

{

- Resource *exposes* Port

where Resource = System, Physical, Software, Organisation, Job or Role

}

AND

connecting the Ports and adding the Interaction Element exchanged. Note that this is the minimum for interface characterisation since the basic interface identification form can be used if only the Interaction Element is known:

{

- Port *from* Port Connection
- Port *to* Port Connection
- Port Connection *exchanges* Interaction Element
- Interaction Element *has part* Interaction Element

AND

{

- Port Connection *realises* Resource Interaction

OR

- Port Connection *uses* Protocol

OR

- Port *implements* Protocol

}

}

Realisation

- Resource Interaction *realises* Need

Optional Tuples

Context - Solution Realises Concept

- Resource *realises* Node

where Resource = System, Physical, Software, Organisation, Job or Role

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

A SV-02 view shall contain:

Interface Identification

- at least 2 Resources connected by a Resource Interaction (one of the resources is the subject of the view) (using Resource Interaction *from* / *to* Resource)
- each Resource Interaction has a unique identifier
- every Resource Interaction must be typed (allowed values = Unknown, Data, Energy or Resource - see TRAK Metamodel document)

Interface Characterisation

- at least 2 Resources ((one of the resources is the subject of the view)
- every Resource has at least 1 Port (using Resource *exposes* Port)
- every Port has at least 1 Port Connection (using Port Connection *from* /*to* Port)
- each Port Connection has a unique identifier

- every Port Connection has at least 1 Interaction Element (using Port Connection *exchanges* Interaction Element)
- every Interaction Element must be typed (allowed values = Unknown, Data, Energy or Resource - see TRAK Metamodel document). Default= 'Unknown' - a possible concern in the architecture description.
- Resource Interaction realised by every Port Connection (Port Connection *realises* Resource Interaction)
- every Port Connection realises a Resource Interaction

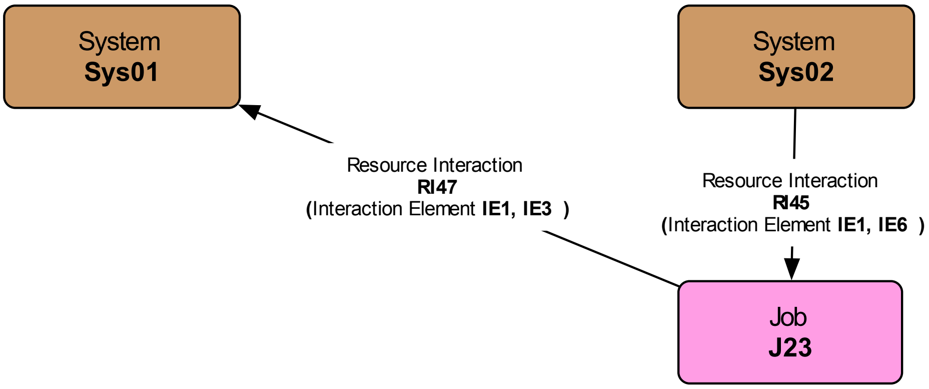
Note:

1. interface identification must be completed before interface characterisation can begin
2. See also TRAK Bye Law BLV-4 in TRAK Architecture Framework document - views for each subject (system of interest)

Presentation Methods

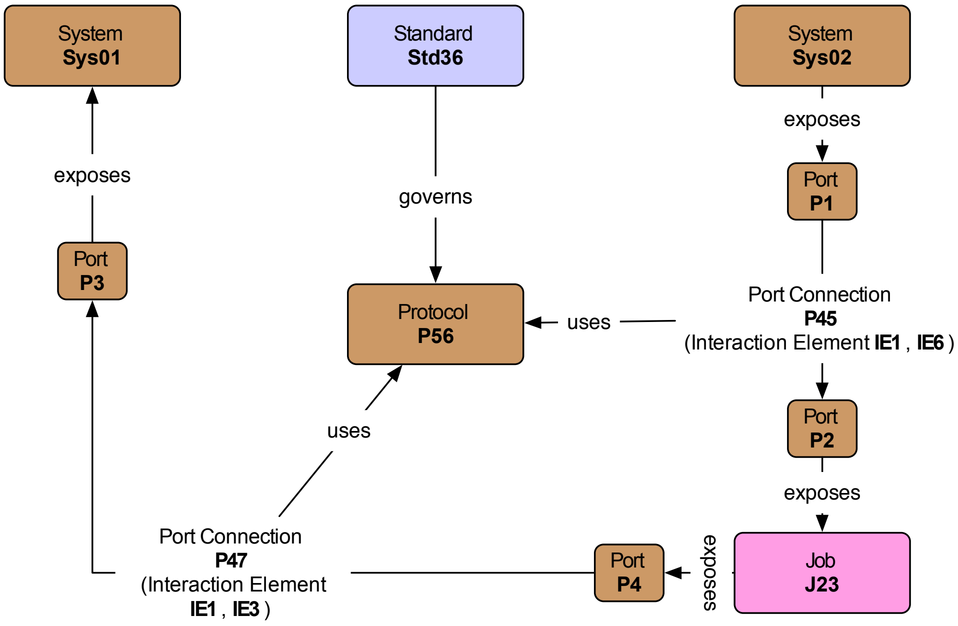
- block diagram (Resource, Port = block, Resource Interaction, Port Connection, TRAK metamodel relationship = line with text label, identifier and direction indicator)

[interface identification]



Note: The resource interactions are typed = 'Data'. If this cannot be attached to the resource interaction(s) an accompanying list of typed resource interactions would be needed.

[interface characterisation]



Note: Assumes interface identification form exists

- table or matrix

column / row headings:

- Resource Interaction identifier
- Resource Interaction Type
- Resource identifier (source)
- Resource identifier (destination)
- Resource Interaction description

- Port Connection identifier
- Port (source)
- Port (destination)
- Interaction Element(s)
- Protocol(s)
- Standard(s)

- N-squared diagram

- Resource identifier (cells on diagonal)
- Resource Interaction identifier - in cell(s) that form intersection(s) between a pair of Resources

- Resource Interaction description - in cell(s) that form intersection(s) between a pair of Resources

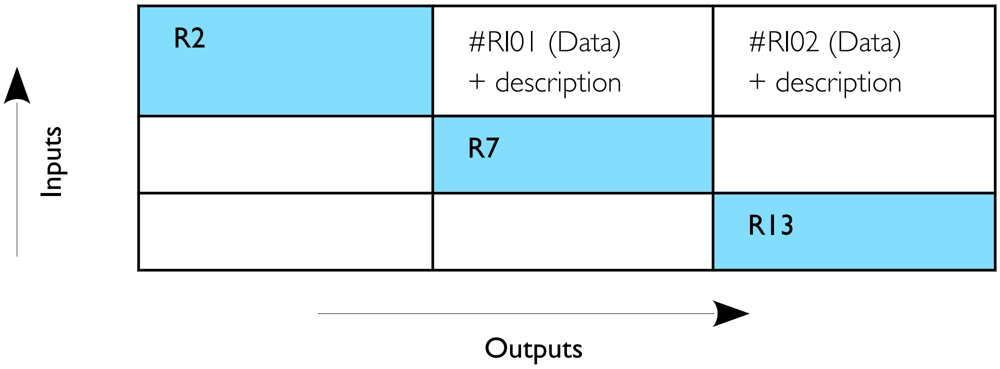
[interface identification]

Resource Interaction Identifier	Source Resource	Destination Resource	Resource Interaction Type	Description
#RI01	R2	R7	Data	nnn mmm
#RI02	R2	R13	Data	mmm sss jj

[interface characterisation]

Resource Interaction Identifier	Port Connection Identifier	Source Resource	Source Port	Dest. Resource	Dest. Port	Interaction Element(s)	Protocol(s)	Description
#RI01		R2		R7				nnn mmm
#RI02		R2		R13				mmm sss jj

- N-squared diagram



R2	#RI01 (Data) + description	#RI02 (Data) + description
	R7	
		R13

Views Needed In Order to Construct

- SV-01 - master architecture view for Resource (Job, Physical, Organisation, Role, Software, System)

See minimum TRAK architecture description view sets.

Consistency Rules

- Resource must appear in SV-01
- **IF** { Role *from* / *to* (Resource Interaction} [Resource] (in SV-02) } **THEN** { Role *extends to* [Resource] (in SV-01) }
- **Topological Realisation of Node : IF** {Node *has* Need (CV-01) } **AND** {Resource Interaction *realises* (same) Need (SV-02) } **THEN** {Resource *realises* Node} must be made on the SV-01
- {Resource Interaction *from* / *to* Resource pair in SV-02} must be consistent with (same) Resource *performs* Function *triggers* Interaction Element *triggers* Function *performed by* (second) Resource in SV-07}
- Interaction Element in {Resource Interaction *from* / *to* Resource *carries* Interaction Element (SV-02)} is same Interaction Element in {(same) Resource *performs* Function *triggers* Interaction Element *triggers* Function *performed by* (second) Resource (in SV-07)}

Comments

The SV-02 is the master architecture view for Interaction Element, Port, Port Connection, Protocol, Resource Interaction.

Note - the level of detail of the SV-02 view, whether identifying or characterising exchanges will, if a functional sequence is established using the SV-07, affect what is shown on a SV-03 Solution Resource Interaction to Function Mapping Views within the architecture description. This is a because the Resource *performs* Function *triggers* Interaction Element *triggers* Function *performed by* (other) Resource on the SV-07 has to be consistent with the Resource Interaction between the two Resources. i.e. if Interaction Element are shown on the SV-02 then they will have to be shown on the SV-03 so that all Interaction Elements exchanged are linked to the Functions they support. If only Resource Interactions have been identified then the SV-03 will only show the Resource Interaction.

SVp-03 Solution Resource Interaction to Function Mapping

Version Number

6

Date

8 December 2017

Description

Maps resource interactions to functions for justification, completeness.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Solution Acquirer, Developer, Maintainer, Operator, Owner, Trainer, User	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?

Table 2-15 SVp-03 Stakeholder Concerns

Anti - Concerns

-

Declared Tuples

Subject

- Resource *performs* Function

where Resource = System, Physical, Software, Organisation, Job or Role

Interface Identification

Identifying the Resource Interaction between a pair of Resources requires 1 of the following tuples:

{{

- Resource – *from* / *to* (Resource Interaction) - Resource

where Resource = System, Physical, Software, Organisation, Job or Role

}

OR

Interface Characterisation

{

- Resource Interaction *carries* Interaction Element
- Interaction Element *has part* Interaction Element

}}

AND

{

- Resource Interaction *supports* Function

}

Optional Tuples

Interface Characterisation

- Resource *exposes* Port
- Port *from/to* Port Connection
- Port Connection *exchanges* Interaction Element
- Interaction Element *triggers* Function
- Function *triggers* Interaction Element
- Port Connection *realises* Resource Interaction

Universal

- Claim *about*, Concern *about, traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

A SV-03 view shall contain:

- at least one Resource (by the subject of the view)

- every Resource must be connected to at least one Function (using Resource *performs* Function)
- at least one Resource Interaction (involving the subject Resource using Resource Interaction *from* / *to* Resource)
- every Resource Interaction must be connected to at least one Function (using Resource Interaction *supports* Function), if not then a Concern *
- every Function must be connected to at least one Resource Interaction (using Resource Interaction *supports* Function) if not then a Concern *

* Note: The purpose of the viewpoint is to identify unmapped resource interactions and unmatched functions. Unmapped items are to be flagged by connecting to a Concern (using Concern *about* Function or Concern *about* Resource Interaction) identifying this concern and reported on in the MV-02. As items are mapped on this view they are disconnected from the Concern.

The SV-03 shall be consistent with the level of detail shown in the corresponding SV-02:

- if the Resource Interactions shown on any SV-02 only identify the Resource Interaction(s) then the SV-03 need only do the same. This simply asserts that a resource interaction supports a function - no understanding of how.
- if the Resource Interactions shown on any SV-02 characterise the Resource Interaction(s) by showing the Interaction Element(s) passed the SV-03 shall also show the Interaction Element(s). This justifies the previous assertion by describing how the function is supported.

Presentation Methods

- table or matrix

Resource Identifier	Function (of that Resource)	Resource Interaction Identifier	Justification
#R1	F1 ...	RI08 ...	
#R2	F2	R109 ...	

Views Needed In Order to Construct

- SV-01 - master architecture view for Resource (Job, Organisation, Physical, Role, Software, System)
- SV-02 - master architecture view for Interaction Element, Port, Port Connection, Resource Interaction
- SV-04 - master architecture view for Function

See minimum TRAK architecture description view sets.

Consistency Rules

Resource *performs* Function must be in Resource Interaction *from* / *to* (same) Resource

The relationship between Function and Resource Interaction established on the SV-03 must be consistent with any established by linking functions via interaction elements when describing a sequence on the SV-07. (see SVp-07 Solution Sequence Viewpoint).

Comments

SVp-04 Solution Function

Version Number

9

Date

8 December 2017

Description

Describes the functions of parts of the solution.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Solution Acquirer, Developer, Maintainer, Operator, Owner, Trainer, User	Have all the solution functions been identified? What does each part do?

Table 2-16 SVp-04 Stakeholder Concerns

Anti - Concerns

-

Declared Tuples

At least:

- Resource *performs* Function

where Resource = System, Physical, Software, Organisation, Job or Role

- Function *has part* Function
- Function *is quantified by* Metric
- Metric *has part* Metric

Optional Tuples

- Function *realises* Concept Activity

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

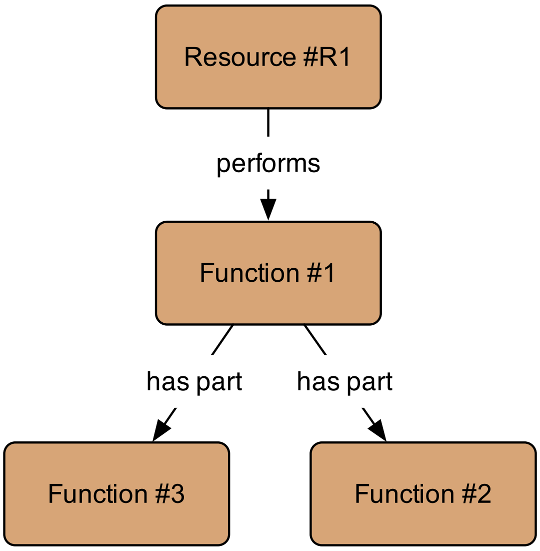
A SV-04 view shall contain:

- at least 1 Resource (the subject of the view)
- the subject Resource must have at least 1 Function (using Resource *performs* Function)

Presentation Methods

- block diagram (Resource, Function = block, TRAK relationship = line with text label and direction indicator)

hierarchical form:



Views Needed In Order to Construct

- SV-01 - master architecture view for Resource

See minimum TRAK architecture description view sets.

Consistency Rules

- Resource must appear in SV-01

- **Functional Realisation of Node.: IF {**Node *conducts* Concept Activity (CV-05)**} AND {**Resource *performs* Function *realises* (same) Concept Activity (SV-04 + SV-05) **} THEN {**Resource *realises* Node**}** must be made on the SV-01.

Comments

The SV-04 is the master architecture view for Function.

If ordering of functions is a concern use the Solution Sequence View (defined by the. SVp-07).

SVp-05 Solution Function to Concept Activity Mapping

Version Number

7

Date

8 December 2017

Description

Maps the solution functions (SV-04) back up to the logical concept activities (defined in the CV-05).

Concerns Addressed

Stakeholder	Concern of Stakeholder
Concept Developer, Disposer, Maintainer, Operator, Owner, Trainer, User **Solution** Acquirer, Developer, Maintainer, Operator, Owner, Trainer, User	Do the solution functions meet all of the concept activities? Is there unwanted solution functionality?

Table 2-17 SVp-05 Stakeholder Concerns

Anti - Concerns

-

Declared Tuples

Defining the solution function:

- Resource *performs* Function

where Resource = System, Physical, Software, Organisation, Job or Role

- Function *has part* Function

Defining the Concept Activity:

- Node *conducts* Concept Activity
- Concept Activity *has part* Concept Activity

Mapping the solution to the concept:

- Function *realises* Concept Activity

Optional Tuples

- Resource *realises* Node

where Resource = System, Physical, Software, Organisation, Job or Role

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

A SV-05 view shall contain:

- at least one Resource (the subject of the view)
- every Resource must be connected to at least one Function (using Resource *performs* Function)
- at least one Node
- every Node must be connected to at least one Concept Activity (using Node *conducts* Concept Activity)
- every Concept Activity must be connected to at least one Function (using Function *realises* Concept Activity), if not then a Concern (see *)
- every Function must be connected to at least one Concept Activity (using Function *realises* Concept Activity), if not then a Concern (see *)

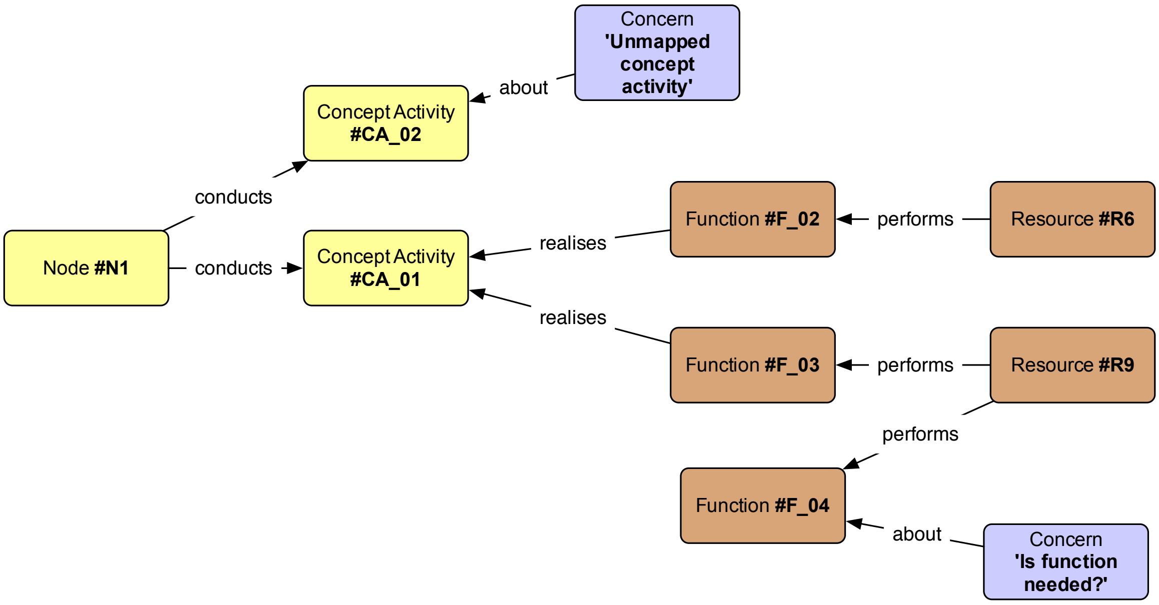
* Note: The purpose of the viewpoint is to identify unmapped functions and unmatched concept activities. Unmapped items are to be flagged by connecting to a Concern (using Concern *about* Concept Activity or Concern *about* Function) identifying this concern and reported on in the MV-02. As items are mapped on this view they are disconnected from the Concern.

Presentation Methods

- table or matrix

Node	Concept Activity	Function	Resource	Description
#N1	CA_02	(unmapped)	(unmapped)	Concern - Unmapped Concept Activity
#N1	CA_01	F_02	#R6	nnn mmm
#N1	CA_01	F_03	#R9	mmm sss jj
(unmapped)	(unmapped)	F-04	#R9	Concern - is function needed?

- block diagram (Node, Concept Activity, Resource, Function = block, TRAK relationship = line with text label and direction indicator)



Note: Any unmapped concept activity or function as far as the subject pair(s) (particular Node and Enterprise pair) being described represents a concern.

Views Needed In Order to Construct

- SV-01 - master architecture view for Resource (Job, Physical, Organisation, Role, Software, System)
- SV-04 - master architecture view for Function
- CV-01 - master architecture view for Node
- CV-05 - master architecture view for Concept Activity

See minimum TRAK architecture description view sets.

Consistency Rules

- Resource must appear in SV-01
- any Node must appear CV-01
- every Node conducts Concept Activity must appear in CV-05
- every Resource *performs* Function must appear in SV-04
- **Functional Realisation of Node.: IF** { Node *conducts* Concept Activity (CV-05) } **AND** { Resource *performs* Function *realises* (same) Concept Activity (SV-04 + SV-05) } **THEN** { Resource *realises* Node } must be made on the SV-01.

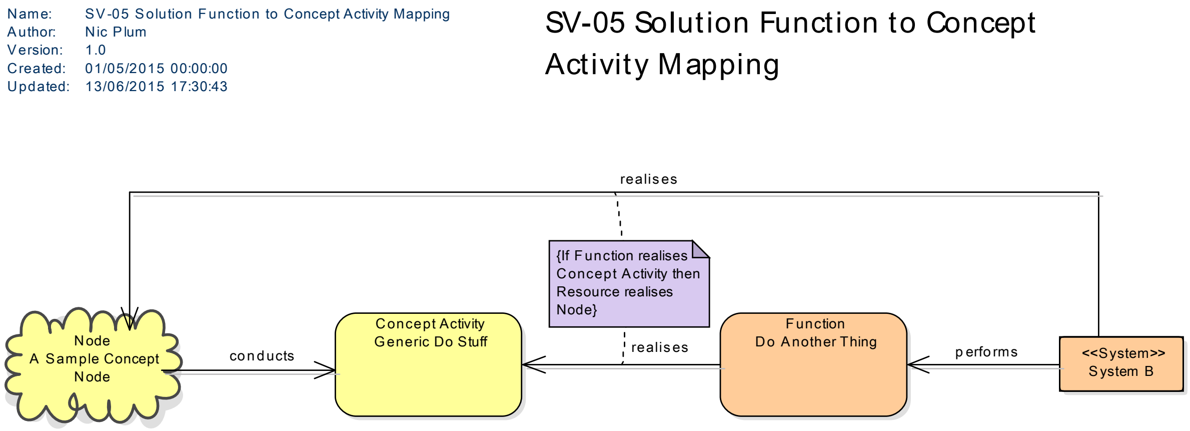

Name: SV-05 Solution Function to Concept Activity Mapping
Author: Nic Plum
Version: 1.0
Created: 01/05/2015 00:00:00
Updated: 13/06/2015 17:30:43
SV-05 Solution Function to Concept Activity Mapping
realises
{If Function realises Concept Activity then Resource realises Node}
Node
A Sample Concept Node
conducts
Concept Activity
Generic Do Stuff
realises
Function
Do Another Thing
performs
<<System>>
System B

Figure 2-1 Consistency - Functional Realisation of Node

Comments

As a mapping view the SV-05 is not a master architecture view.

SVp-06 Solution Competence

Version Number

8

Date

8 December 2017

Description

Describes the competence needed for a role. Justified by linking to function or the extent of the resource to which the role applies.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Solution Acquirer, Builder, Developer, Maintainer, Operator, Owner, Trainer, User	Does the organisation or job through its role have the necessary competence to conduct the function? Is the competence consistent with the solution?

Table 2-18 SVp-06 Stakeholder Concerns

Anti- Concerns

-

Declared Tuples

Role-player:

{

- Organisation *plays* Role

OR

- Job *plays* Role

}

AND

Needing competence:

- Role *requires* Competence

Rationale:

- Competence *to conduct* Function

See Consistency Rules.

Optional Tuples

Context - Structural

- Organisation *has part* Organisation
- Organisation *has part* Job
- System *is configured with* Organisation
- System *is configured with* Job
- System *is configured with* Role

Context - Role Extent

- Role *extends to* Resource

where Resource = System, Physical, Software, Organisation, Job or Role

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

A SV-06 view shall contain:

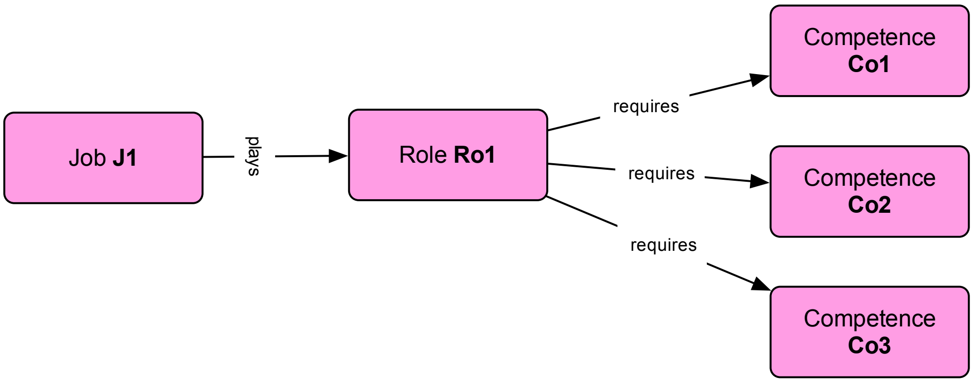
- at least 1 Organisation or Job (the subject of the view)
- the subject Organisation / Job must be connected to at least 1 Role (subject role) (using Organisation / Job *plays* Role)
- the subject Role must be connected to at least 1 Competence (using Role *requires* Competence)

If the Function which requires this Competence is already in a SV-04 then:

- Competence must be linked to the Function (using Competence *to conduct* Function)

Presentation Methods

- block diagram (Organisation, Job, Role, Competence, Function = block, TRAK relationship = line with text label and direction indicator)



- table or matrix (row/column title = Job, Role, Competence, Function, Description)

Job	Role	Competence	Function	Description
#J1	Ro1	Co1	F2	nnn mmm
#J1	Ro1	Co2	F2	mmm sss jj
#J1	Ro1	Co3	F23	nn sjj wmm

Views Needed In Order to Construct

- SV-01 - master architecture view for Resource (Job, Physical, Organisation, Role, Software, System)

See minimum TRAK architecture description view sets.

Consistency Rules

- Resource must appear in SV-01
- Function must appear in SV-04
- 'Role *extends to* Resource' must appear in SV-01
- 'System *is configured with* Role *requires* Competence *to conduct'* must be consistent with 'System *performs* Function' (in SV-04)
- 'Role *requires* Competence *to conduct* Function' must be consistent with 'Role *performs* Function' (in SV-04)

- If a suitable function is shown in the SV-04 then the 'Competence *to conduct* Function' tuple must be made in the SV-06 - this is why the an 'optional tuple' is shown in the Declared Tuples section. It guarantees coverage of this tuple in the TRAK metamodel.

Comments

The SV-06 is the master architecture view for competence.

SVp-07 Solution Sequence

Version Number

11

Date

8 December 2017

Description

Describes the order in which functions or events occur.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Solution Acquirer, Developer, Maintainer, Operator, Owner, Trainer, User	In what order do things need to happen?

Table 2-19 SVp-07 Stakeholder Concerns

Anti - Concerns

-

Declared Tuples

Functional Sequence

- Resource (System or Physical or Software or Organisation or Job or Role) *performs* Function
- Function *precedes* Function
- Function *triggers* Interaction Element
- Interaction Element *triggers* Function

Interaction Sequence

- Resource Interaction *from* / *to* Resource
- Resource Interaction *carries* Interaction Element

Optional Tuples

Context

To establish what 'thing' is performing functional or interaction sequence and where it sits in the bigger picture:

- System *is configured with* Resource
- Organisation *has part* Job
- Organisation *has part* Organisation
- Organisation *is member of* Organisation
- Software *has part* Software
- Job *plays* Role
- Organisation *plays* Role
- Function *has part* Function
- Interaction Element *has part* Interaction Element

where Resource = System, Physical, Software, Organisation, Job or Role

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

A SV-07 view shall contain:

Function Sequence

- at least one Resource (the subject of the view)
- every Resource must perform at least one Function (Resource *performs* Function)
- every Function must be connected to another Function by **EITHER** (Function *triggers* Item *triggers* Function) **OR** (Function *precedes* Function)

Interaction Sequence

- at least 2 Resources (the subject of the view)
- each Resource pair is connected by a Resource Interaction (using Resource Interaction *from* / *to* Resource)

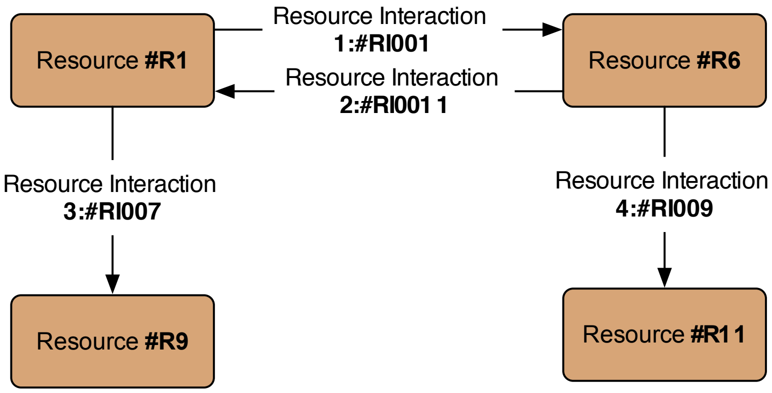
General

- a means of identifying the order in which all Functions/Resource Interactions occur e.g. by numbering or explicit layout on a time line

Presentation Methods

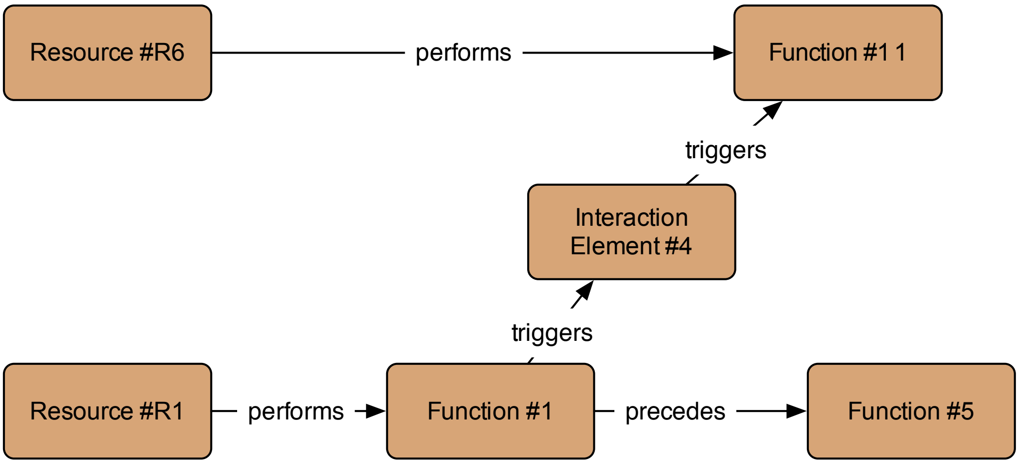
- sequence diagram
- communication diagram

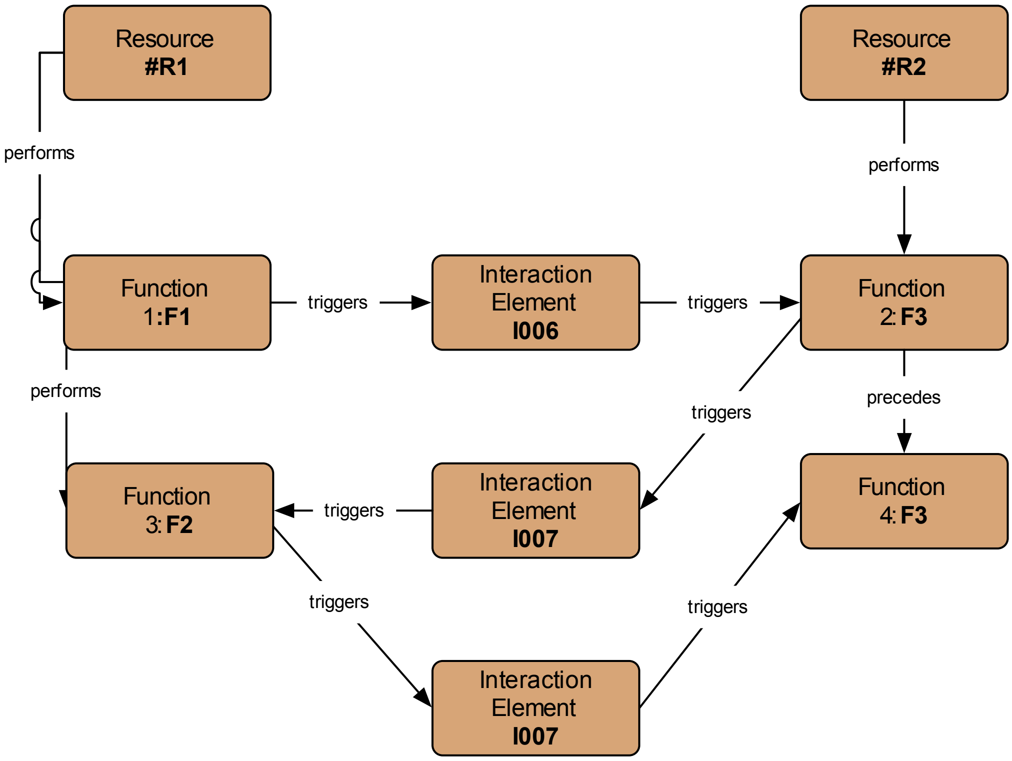
[Resource interaction sequence only, no interaction elements shown]



Note: labelling shown is of the form: sequence identifier:Resource Interaction Identifier.

Functional sequence:





Views Needed In Order to Construct

- SV-01 - master architecture view for Resource (Job, Physical, Organisation, Role, Software, System)
- SV-04 - master architecture view for Function

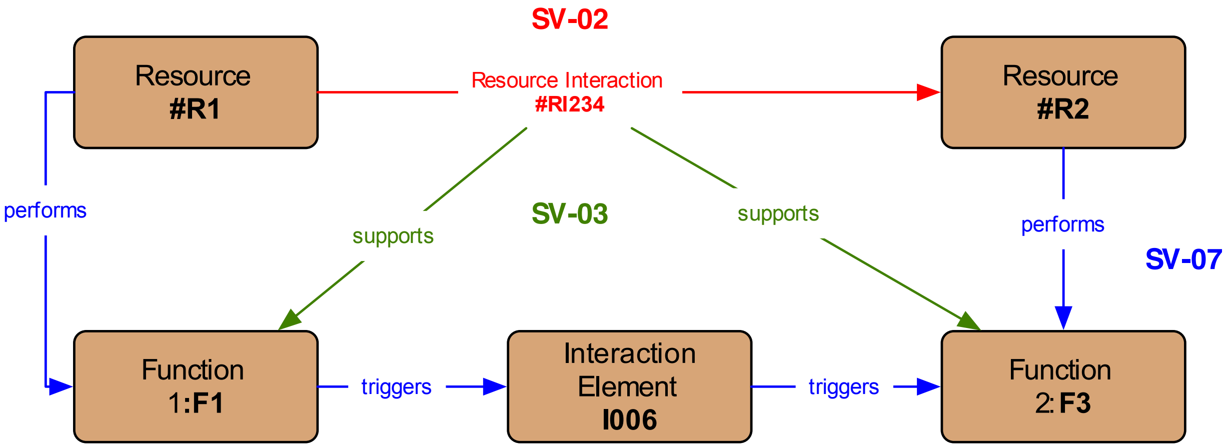
See minimum TRAK architecture description view sets.

Consistency Rules

- any Interaction Elements and Resource Interactions needed to trigger Functions must be consistent with those defined on the SV-02 i.e.
  - **IF** (Resource *performs* Function *triggers* Interaction Element on SV-07) **THEN** (Resource Interaction *from* (same) Resource *carries* (same) Interaction Element on SV-02)
  - **IF** (Interaction Element *triggers* Function *performed by* Resource SV-07) **THEN** (Resource Interaction *to* (same) Resource *carries* (same) Interaction Element on SV-02)

- any Functions shown must be consistent with those defined on the SV-04:
  - **IF** Resource *performs* Function on SV-07 **THEN** ((same) Resource *performs* (same) Function on SV-04)

- any linking of Resources using a Function and an Interaction Element to establish a sequence causes a SV-02 and SV-03 to be created if not already present :
  - Functional justification of resource interaction.: **IF {** Resource *performs* Function *triggers* Interaction Element (SV-07)} AND **{**(second) Resource *performs* (second) Function *triggers* (same) *Interaction Element* (SV-07) **}** **THEN {{**Resource Interaction *from* Resource *to* (second) Resource must be made in SV-02 **}** **AND {**(same) Resource Interaction *supports* Function and (same) Resource Interaction *supports* (second) Function must be made in SV-03 **}}** .



Step 1 :SV-07 IF (R1 linked to R2 by Functions F1, F2 and Interaction Item I006) **THEN**
Step 2 :SV-02 there is a Resource Interaction from R1 to R2
Step 3 :SV-03 Resource Interaction #R1234 supports (Function F1 AND Function F3)

Comments

The Interaction Sequence form requires 2 Resources because 'Function *triggers* Interaction Element' implies that there is another Resource with which that Interaction Element is exchanged.

SVp-11 Solution Event Causes

Version Number

3

Date

20 August 2019

Description

Describes causal or contributory relationships between events that are associated with the system of interest either owing to an error within the system or direct action from something external to the system.

Typically used to represent:

- fault trees for safety to identify a causal chain of events that lead to a set of top ‘feared events’ which need to be managed
- reliability analyses that provide the basis for predicting how reliable (dependable) a system is in delivering its intended behaviour or level of performance.
- the architecture within a solution that makes it robust in the presence of behaviours or inputs that affect its ability to deliver its intended behaviour or performance (diversity, redundancy, fault-tolerance etc.)

The events are typically associated with a failure to follow specified behaviour or a failure to anticipate or specify all the required behaviour as opposed to the SVp-04 Solution Function viewpoint which describes specified (normal) behaviour. Events may be caused by the system of interest or by external systems or 'actors'. This viewpoint therefore provides a means to anticipate what happens when things go wrong.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Solution Acquirer, Developer, Maintainer, Operator, Owner, Trainer, User	How robust is the system to unwanted events? How dependable is the system? What causes events?

Table 2-20 SVp-11 Stakeholder Concerns

The concerns addressed by this viewpoint are:

- how robust is the system to unwanted events?
- how dependable is the system?

- what causes (feared) events?

Anti - Concerns

- Intended or specified function (behaviour) - this is described using the SVp-04 Solution Function viewpoint

Declared Tuples

- Event *caused by* Event
- Event *caused by* Resource
- Event *caused by* Function
- Event *impacts on* Resource
- Event *impacts on* Function

Optional Tuples

- Event *~~poses~~can lead to exposure to* ~~Risk~~Threat

Context – Containing System

Identifying containing system using path consisting of a combination of:

- System *is configured with* Resource
- Software *hosted on* Physical
- Physical *contains* System
- Physical *has part* Physical
- Software *has part* Software
- Organisation *is member of* Organisational
- Organisation *has part* Organisational
- Organisation *has part* Job
- Job *plays* Role
- Organisation *plays* Role
- Resource *performs* Function

Universal

- Claim *about*, Concern *about, traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

A SV-11 shall contain:

- at least one (the subject) Event
- the subject Event is linked to at least one Resource using either 'Event *caused by* Resource' or 'Event *impacts on* Resource'
- the subject Event is linked to one or more events that cause the subject Event. If a combination of events are needed to cause the event this shall be stated using a boolean logical operator (e.g. AND, OR, NOT, XOR or any combination).
- **IF** a Function is linked to an Event using **EITHER** (Event *impacts on* Function **OR** Event *is caused by* Function) **THEN** the associated Resource shall be shown using Resource *performs* Function.

Presentation Methods

- graphical, showing a tree depicting the causal relationships e.g. a fault tree, visualisation of tuples.
- textual e.g. set of tuples as assertions.

Event. Hazardous Radiation Dose Exposure caused by System. Linear Accelerator

Event. Hazardous Radiation Dose Exposure caused by ((Event. Dose set too high AND Event. Alarm not activated)) OR (Event. Dose set correctly AND Event. Radiation))

Figure 2-2 SVp-11 Example - Text

Since a boolean logic operator is dimensionless the combination Event AND/OR/etc. Event is itself an Event.

Views Needed In Order to Construct

Will usually draw upon the following:

- SV-01 Solution Structure View – to identify the Resources involved
- SV-04 Solution Function View – to identify any Functions associated with Events
- SV-13 Solution Risk View – to identify how any Risks that arise are managed.

See minimum TRAK architecture view sets.

Consistency Rules

- any relationship with Function must be consistent with the Resource performing the Function:-
    - IF Event *caused by* Function **AND** Resource *performs* (same) Function **THEN** Event *caused by* (same) Resource.
    - IF Event *impacts on* Function **AND** Resource *performs* (same) Function **THEN** Event *impacts on* (same) Resource.

Comments

The SV-11 is the master architecture view for Event.

SVp-13 Solution Risk

Version Number

4

Date

20 August 2019

Description

Describes the threats posed to a system as a result of vulnerabilities that expose the system of interest (or other resources) to risk. Describes how these are managed, mitigated or controlled so that the risks are kept at a tolerable level.

Typically used to represent:

- how risks are managed, mitigated and controlled, for example by design
- the origins of a risk in terms of particular threats which exploit system vulnerabilities, for example to support an analysis of the security features of a system
- how threats can cause particular events (which might be part of a sequence that leads to a top level event that needs to be prevented, mitigated or minimised - addressed in the SVp-11 Solution Event Causes Viewpoint).

Concerns Addressed

Stakeholder	Concern of Stakeholder
Solution Acquirer, Developer, Maintainer, Operator, Owner, User, Trainer	What threats is the system exposed to? How are the threats mitigated or controlled? What are the vulnerabilities of the system? What are the risks posed to the system or a third party? How does the solution design address vulnerabilities, threats and risks?

Table 2-21 SVp-13 Stakeholder Concerns

The concerns addressed by this viewpoint are:

- 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?

Anti - Concerns

-

Declared Tuples

Identification

- Resource *exposed to* Risk
- Resource *poses* Threat
- Function *poses* Threat (syn. hazard)
- Threat (syn. Hazard) *poses* Risk
- Threat (syn. Hazard) *to* Resource
- Threat (syn. Hazard) *to* Function
- Threat (syn. Hazard) *to* Resource Interaction
- Threat (syn. Hazard) *to* Interaction Element
- ~~Threat (syn. Hazard) *to* Function~~

Analysis

As Identification +

- Resource *has* Vulnerability
- Function *has* Vulnerability
- Resource Interaction *has* Vulnerability
- Interaction Element *has* Vulnerability
- Vulnerability *contributes to* Vulnerability
- Threat (syn. Hazard) *exploits* Vulnerability
- Vulnerability *results in* Risk

Management & Control

As identification +

- Risk *is managed by* Mitigation
- Mitigation *uses* Resource
- Mitigation *uses* Function

Optional Tuples

Context – Containing System

Identifying containing system using path consisting of a combination of:

- System *is configured with* Resource
- Software *hosted on* Physical
- Physical *contains* System
- Physical *has part* Physical
- Software *has part* Software
- Organisation *is member of* Organisational
- Organisation *has part* Organisational
- Organisation *has part* Job
- Job *plays* Role
- Organisation *plays* Role

Context – Events

- Event *can lead to exposure to* ~~Risk~~Threat
- Event *caused by* Event

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

A SV-13 shall contain:

Identification

- at least one (the subject) Resource
- the subject Resource is linked to at least one Risk using 'Resource *exposed to* Risk'

- every Risk is linked to at least one Threat using 'Threat *poses* Risk'
- every Threat is linked to the subject Resource using
    - EITHER 'Threat *to* Resource'
    - OR ('Threat *to* Function' AND '(subject) 'Resource *performs* (same) Function')
    - OR ('Threat *to* Resource Interaction' AND 'Resource Interaction *from* / *to* Resource' where one of the Resources is the subject Resource)
    - OR ('Threat *to* Interaction Element' AND 'Port Connection *from* / *to* Port' AND 'Resource *exposes* Port' where one of the Resources is the subject Resource)

i.e there must at least one Resource - Risk - Threat - (same) Resource path

Analysis

As Identification +

- every Threat is linked to at least one (subject Resource) Vulnerability using 'Threat *exploits* Vulnerability'
- every (subject Resource)Vulnerability is linked to the (subject) Resource using 'Resource *has* Vulnerability'
- every (subject Resource) Vulnerability is linked to at least one Risk using 'Vulnerability *results in* Risk'

Management & Control

As Identification +

- every Risk is linked to at least one Mitigation using 'Risk *is managed using* Mitigation'
- every Mitigation is linked to the means of mitigation in the solution using 'Mitigation *uses* Resource' OR 'Mitigation *uses* Function AND 'Resource *performs* Function'

Presentation Methods

- graphical, showing a tree depicting the causal relationships e.g. a fault tree, visualisation of tuples.

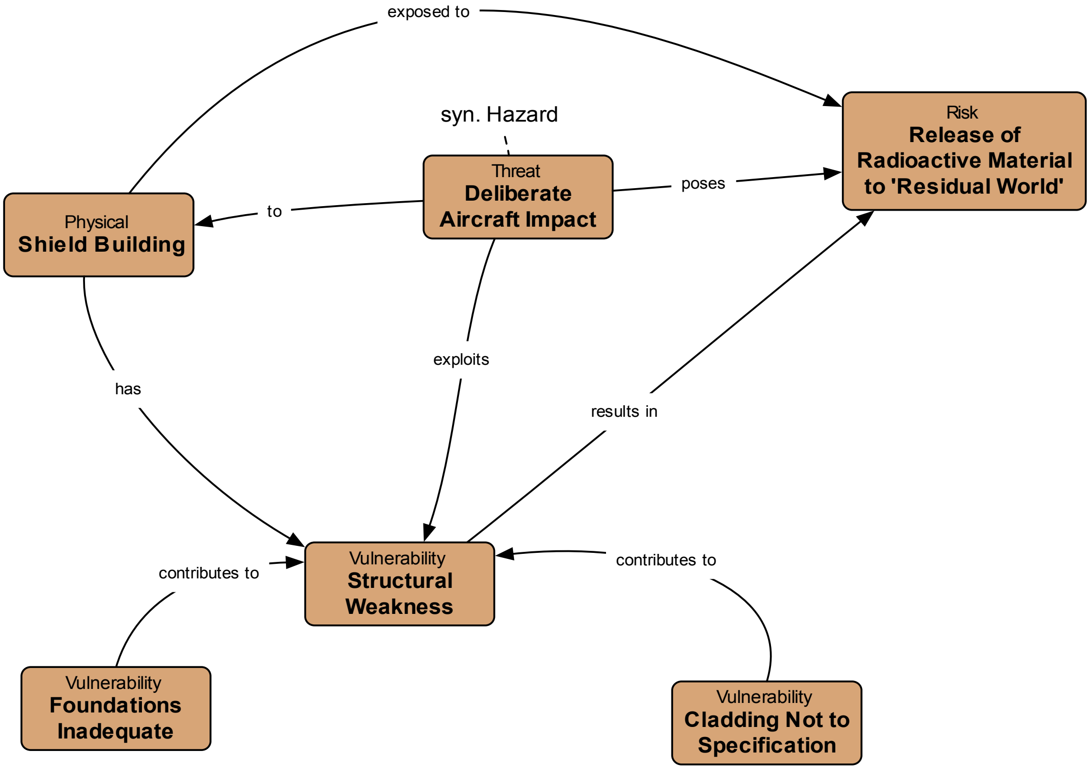


Figure 2-3 SVp-13 Example

- textual e.g. set of tuples as assertions.

Views Needed In Order to Construct

Will usually draw upon the following:

- SV-01 Solution Structure View - to identify the Resources involved
- SV-02 Solution Resource Interaction View – if any Resource Interactions or Interaction Elements are involved
- SV-04 Solution Function View- if any Functions are involved

See minimum TRAK architecture view sets.

Consistency Rules

Comments

The SV-13 is the master architecture view for Mitigation, Risk, Threat and Vulnerability.

MVp-01 Architecture Description Dictionary

Version Number

6

Date

8 December 2017

Description

Defines each element used in the architecture description. This is used to explain to others what each element is intended to represent and is necessary in preserving the meaning. It is also necessary if an element is to be reused correctly and consistently.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Enterprise: Developer, Maintainer , Owner **Concept**: Developer, Maintainer, Owner, User **Solution**: Acquirer, Developer, Disposer, Maintainer , Operator, Owner, Trainer, User **Architecture Task**: Owner, User	Is the architecture description portable? Can it be understood in the way it was intended to be?

Table 2-22 MVp-01 Stakeholder Concerns

Note that the potential set of stakeholders for this view is large because it not only involves the enterprise, concept and solution but the architecture task and lay readers of the architecture description (users of the architecture task).

Anti-Concerns

-

Declared Tuples

Generally the view will be formed from individual elements. There are special cases, for example when importing and connecting a third party architecture description when the following may be used:-

- Architecture Description Element *is a* Architecture Description Element
- Architecture Description Element *equivalent to* Architecture Description Element

Optional Tuples

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

A MV-01 view shall contain:

- every element used in the architecture description
- each element must have a unique means of identification (otherwise the description/definition can't be tied to the element)
- a description or definition for every architecture description element
- any 'is a' relationship only connects a non-TRAK metamodel element to a TRAK metamodel element [any taxonomy hierarchy of TRAK elements needs to be declared in its respective Master View with other similar elements – see Table 3-1 Master Architecture View for Each TRAK Metamodel Element].
- Any '*equivalent to*' relationship is consistent with the definition of all the TRAK metamodel participating in the relationship [e.g. a TRAK::System element cannot be equated to a method].

Presentation Methods

- table

Identifier (if name not unique)	Name	Description / Definition
1	mmmm	n cbv shw jjs
3	bbbbb	vvs ajhjw oa

Views Needed In Order to Construct

- the set of all the views within the AD (since the TRAK Bye Laws require every architecture tuple within the AD to appear on a view.

See minimum TRAK architecture description view sets.

Consistency Rules

Comments

Easiest method of production is often a database query if the architecture modelling tool supports this. It does, however, assume that the architect has provided a description or definition for each architecture description element!

MVp-02 Architecture Description Design Record

Version Number

14

Date

7 February 2018

Description

Describes the purpose, scope and extent of the architecture task and the architecture description. Describes any findings that arose from the architecture modelling.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Enterprise Builder, Developer, Maintainer, Owner, **Architecture Task** Builder, Developer, Owner, User	Do we understand the scope of the architectural task? What are the issues and findings that resulted?
Note: It is likely that the developer and builder roles of the architecture task are likely to be performed by the same person.	

Table 2-23 MVp-02 Stakeholder Concerns

Anti - Concerns

-

Declared Tuples

ISO 42010 AD Scope

Defining the scope of the architecture description / task:

- Contract / Standard / Requirement *governs* Architecture Task [to allow version of framework, metamodel etc to be declared]
- Architecture Task *traces to* Requirement
- Architecture Task *has part* Architecture Task
- Architecture Task *delivers* Architecture Product (Architecture Description, Architecture View)

- Organisation or Job *has* Concern
- Concern *about* Architecture Description Element
- Contract/Standard/Requirement *governs* Architecture Description Element
- Architecture Description Element *satisfies* Contract/Standard/Requirement

Usually the following stakeholders will take at least the role of 'Sponsor' although other roles might be relevant to the architectural task:

- Organisation *plays* Role
- Job *plays* Role
- Role *extends to* Architecture Task

e.g. ÜberChief:Job *plays* Sponsor:Role *extends to* myTask:Architecture Task

Describing how the architecture description addresses the stakeholder concerns:

- Architecture Description *has part* Architecture Description
- Architecture Description *addresses* Concern
- Architecture View *addresses* Concern
- Architecture Description *has part* Architecture View

and reference documents included:

- Architecture Description Element *traces to* Document / Contract / Standard / Architecture View / Architecture Description
- Document *issued by* Organisation
- Document *has part* Document

Architecture Task Findings

ISO 42010 AD Scope +

- Concern *about* Architecture Description Element – these will be new concerns arising during the task (pre-existing ones addressed under ISO 42010 Scope)

Optional Tuples

- Organisation *governs* Project
- Architecture Description Element *traces to* Argument
- Architecture Description Element *traces to* Contract
- Architecture Description Element *traces to* Document

- Architecture Description Element *traces to* Requirement
- Architecture Description Element *traces to* Standard
- Claim *about* Architecture Description Element
- Requirement *governs* Architecture Product (Architecture Description, Architecture View)
- Standard *governs* Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

ISO 42010 AD Scope

A MV-02 view shall contain:

- at least one stakeholder (represented using Organisation/Job *plays* Role of 'Stakeholder')
- at least one architecture task ("the task")
- Role of Stakeholder is connected to "the task" (Role *extends to* Architecture Task)
- every stakeholder has at least 1 concern (Role of 'Stakeholder' *has* Concern - stakeholder concern)
- every stakeholder Concern is related to at least one thing (Concern *about* Architecture Description Element)
- every stakeholder concern is addressed by at least one View (using Architecture View *addresses* Concern)
- references to any task definition documents (Architecture Description Element *traces to* Document)
- references to any normative documents - relevant to the task execution (Contract / Standard *governs* Architecture Description Element) including the version of TRAK, for example.

Note: Initially architecture views won't exist and can be identified by placeholders. These can be replaced by the actual view names or links to the views produced within the architecture description for the task.

Architecture Task Findings

A MV-02 view describing the ISO 42010 AD Scope must exist.

In addition one or more MV-02 views must contain:

- concerns that arise from the task or the analysis (Concern *about* Architecture Description Element)
- each Concern is related to at least one thing (Concern *about* Architecture Description Element)

Note that the architecture description will itself contain views that, say, contrast the current and proposed architectures or which identify gaps or shortfalls. In this case the MV-02 must make reference to these.

ISO 42010 AD Documentation

Irrespective of any other uses of the MV-02 the following must be provided within the AD (Model):

- AD version identifier e.g. number, date, time
- task sponsor
- task scope
- concerns addressed
- assumptions & limitations made
- information sources used
- architect(s)/modeller(s) involved
- findings (concerns arising, observations, conclusions) i
- any decisions affecting the architecture being described, the rationale behind the decisions, the elements affected, the consequences, any supporting information
- models used & developed for task
- any model dependencies

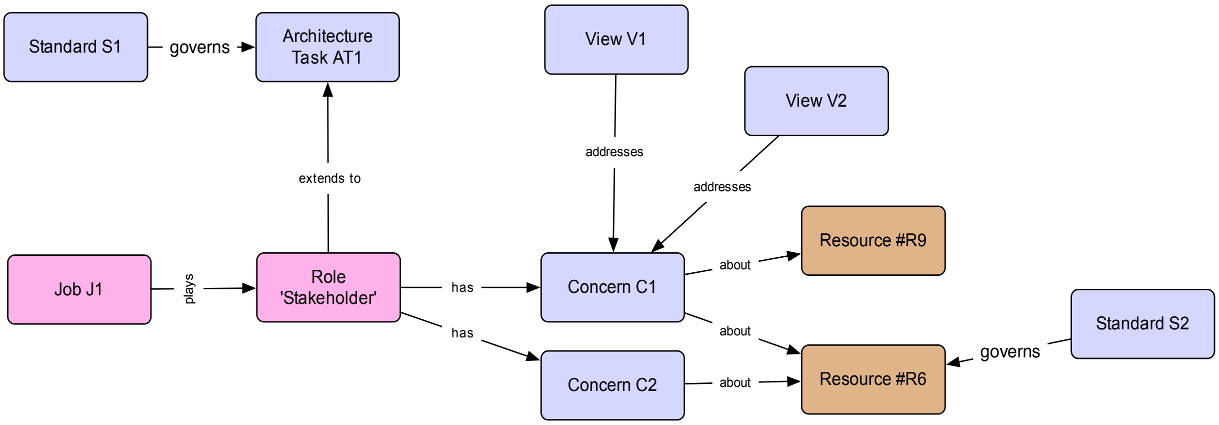
The MV-02 must at the very least reference all the above information.

Presentation Methods

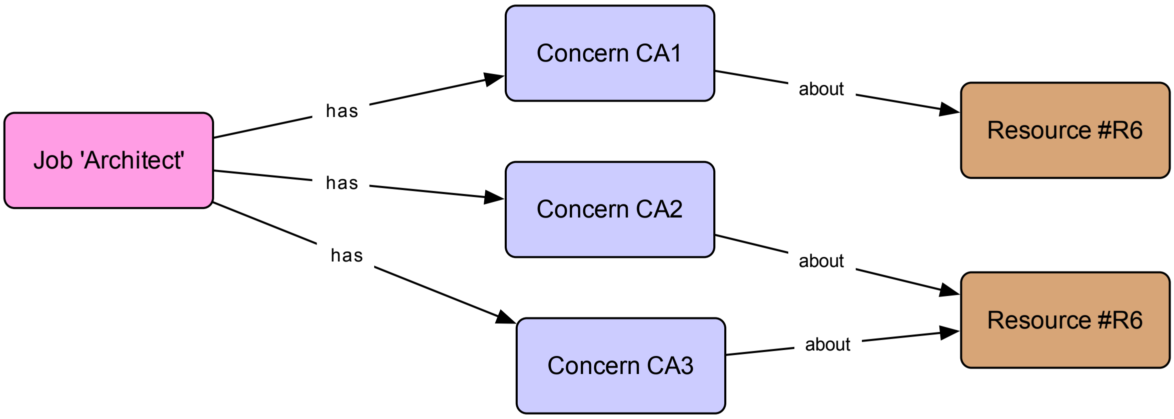
Likely to be a mixture of:

- Text document(s). Some modelling tools will allow document-like formatted text to be embedded or attached to architectural elements within the AD.
- block diagram (Architecture Task, Resource, View, Concern, Requirement, Standard, Document = block, TRAK relationship = line with text label and direction indicator)

ISO 42010 AD Scope



Findings



Can be used to list the architect's concerns as a result of analysis during the task.The concerns are linked to the job of 'architect' in this MV-02 to distinguish them from the stakeholder's originating concerns.

Views Needed In Order to Construct

- SV-01 - master architecture view for Role, Job, Organisation (used to describe the role of sponsor) and usually (different) Resource (which the concern relates to)

See minimum TRAK architecture description view sets.

Consistency Rules

Comments

The MV-02 can be used in several different ways:

1. As the Master Architecture view for Concern it collects together all the concerns expressed in the architecture description (model).
2. To record/capture the nub of the discussions with the sponsor for the task.

3. In conjunction with a package diagram it can be used (after 2) to plan what models are needed for the task.
4. To outline the views that present the results and thereby provide directed points of navigation into the other views within the architecture description.
5. To help document the development of the architecture description for a design record.
6. To help document considerations that affect or might affect the portability of the architecture description (in conjunction with the MV-01).

The MV-02 is the master architecture view for Architecture Description, Architecture Task, Architecture View, Concern, Document.

MVp-03 Requirements & Standards

Version Number

10

Date

8 December 2017

Description

Describes the constraints that apply to an architecture element through requirements and standards or how standards depend on one another.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Acquirer, Operator, User, Trainer, Maintainer, Regulator, Auditor, Builder, Developer, Owner	What other constraints / requirements through normative documents / standards apply (or will apply) to the enterprise, concept, procurement, solution or architecture task?
Note: Any of these roles may be applied to the Enterprise, Concept, Procurement, Solution or Architecture Task	

Table 2-24 MVp-03 Stakeholder Concerns

Anti - Concerns

-

Declared Tuples

The 3 TRAK metamodel elements that represent a requirement or a constraint are Contract, Requirement (atomic or individual) and Standard, both of which can be applied to any architecture description element. Standard and Contract are both normative documents.

These represent any requirements/constraints applied to any element of the architecture being described or the architecture description or the task itself:

- Contract / Requirement / Standard *governs* Architecture Description Element
- Architecture Description Element *satisfies* Contract / Requirement / Standard
- Architecture Description Element *traces to* Contract / Requirement / Standard

- Contract *applies* Standard
- Contract *depends on* Contract
- Contract *supersedes* Contract
- Contract *has part* Contract
- Contract *has part* Requirement
- Standard *has part* Requirement
- Standard *depends on* Standard
- Standard *enacts* Standard
- Standard *equivalent to* Standard
- Standard *applies* Standard
- Standard *supersedes* Standard
- Standard *has part* Standard
- Requirement *has part* Requirement
- Requirement *derived from* Requirement
- Contract / Standard *issued by* Organisation

Optional Tuples

- Claim *about* Architecture Description Element
- Architecture Description Element *traces to* Argument
- Architecture Description Element *traces to* Document
- Document *has part* Document

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

The MV-03 view must contain:

- at least one (Contract or Requirement or Standard)
- every Requirement must be connected to at least one other Architecture Description Element using
  - Requirement *governs* Architecture Description Element
  - Architecture Description Element *satisfies* / *traces to* Requirement
  - Requirement *derived from* / *has part* Requirement

  - Contract / Standard *has part* Requirement

- every Standard must be connected to at least one other Architecture Description Element using
  - Standard *governs* Architecture Description Element
  - Contract *applies* Standard
  - Architecture Description Element *satisfies* / *traces to* Standard
  - Standard *issued by* Organisation
  - Standard *has part* / *depends on* / *supersedes* / *enacts* / *equivalent to* / *applies* Standard
  - Standard *has part* Requirement
- every Contract must be connected to at least one other Architecture Description Element using
  - Contract *governs* Architecture Description Element
  - Contract *applies* Standard
  - Architecture Description Element *satisfies* / *traces to* Contract
  - Contract *issued by* Organisation
  - Contract *has part* / *depends on* / *supersedes* Contract
  - Contract *has part* Requirement
- every Architecture Description Element must be uniquely identified
- Note:

  If no issue or version number for Standard is identified it is assumed that the latest version applies (even if subsequently withdrawn).This allows an architect to declare that a standard applies before they've identified the version.

- attributes should be used to add appropriate detail (TRAK Metamodel document)
  - Requirement has attributes that include requirement identifier, sequence identifier (to describe an order of requirements such as in a document), compliance level (e.g. mandatory or desirable), priority, paragraph number, type (e.g. legal, commercial or technical)
  - Architecture Description Element has attributes that include security descriptors
  - Document has attributes that include Dublin Core Metadata elements
  - Standard has attributes that include issue date, part, number and withdrawal date

Presentation Methods

- block diagram

Constraints

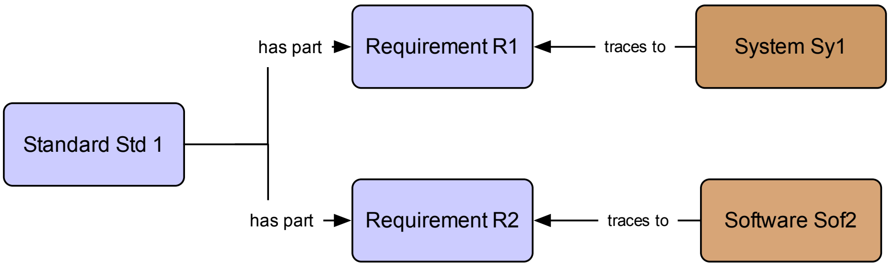


Figure 2-4-MV-03 Example 1 - Requirement Trace to Standard

Note: A Standard can be used to represent a requirement document (collection of requirements).

Contract Standards Assessment

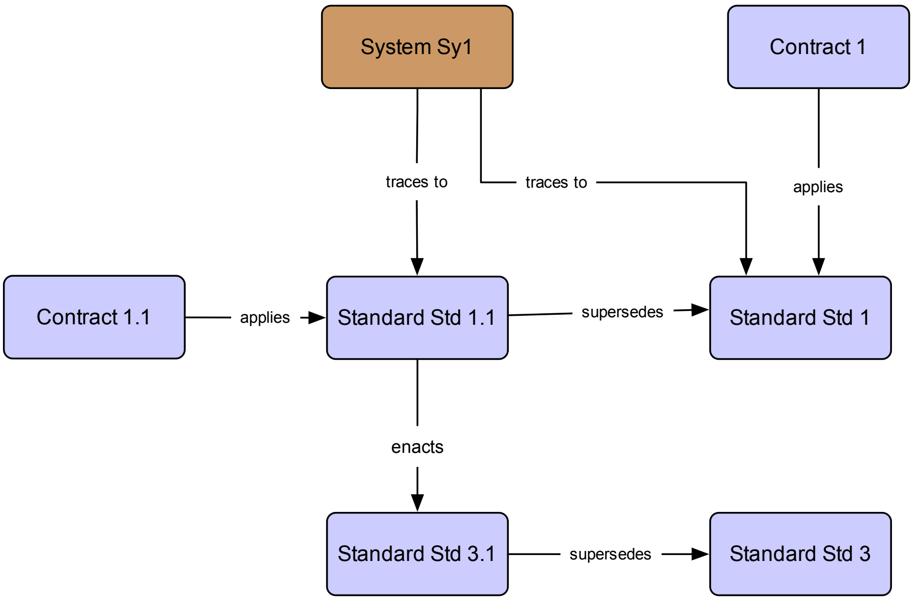


Figure 2-5-MV-03 Example 2 - Contract Standards Assessment

Note: Contract has attributes for start and finish dates and parties to the contract. See TRAK Metamodel document.

- table

Standard (identifier and version)	Architecture Description Element governed	Comments
ISO/IEC 42010:2007	mmmm	n cbv shw jjs
RFC4677	bbbbb	vvs ajhjw oa

Views Needed In Order to Construct

The MV-03 will usually, but not always, be constructed after other views. Depending on the area in which the constraints apply the following views will precede the MV-03 as they provide the 'things' to which the constraints attach to.

Capability constraints

- EV-01 - master architecture view for Enterprise, Enterprise Goal
- EV-02 - master architecture view for Capability

Concept Constraints

- CV-01 - master architecture view for Node, Need
- CV-03 - master architecture view for Item, Item Exchange
- CV-05 - master architecture view for Concept Activity

Procurement Constraints

- PrV-01 - master architecture view for Project
- PrV-02 - master architecture view for Milestone, Project Activity

Solution Constraints

- SV-01 - master architecture view for Resource (Job, Organisation, Physical, Role, Software, System)
- SV-02 - master architecture view for Interaction Element, Port, Port Connection, Protocol, Resource Interaction
- SV-04 - master architecture view for Function
- SV-06 - master architecture view for Competence

- SV-11 – master architecture view for Event
- SV-13 – master architecture view for Mitigation, Risk, Threat and Vulnerability

Management Constraints

- MV-02 - master architecture view for Architecture Description, Architecture Task, Architecture View, Concern, Document.
- MV-03 - master architecture view for Contract, Requirement, Standard.

See minimum TRAK architecture description view sets.

Consistency Rules

Comments

The MV-03 is the master architecture view for Contract, Requirement, Standard.

The views produced are expected to focus primarily on contracts, standards and requirements. Requirement-focussed views will make it possible to show how the architecture description links to products from requirement management tools such as DOORS and also act as a justification for the way in which the architecture has been represented in the architecture description. It is not the purpose of an architecture framework to manage requirements. Is it the purpose of TRAK to provide a means of integrating architecture description with dedicated requirement management tools.

IThe MV-03 can be used to describe Capability Requirement Documents, Operational Requirement Documents and System Requirement Documents. Requirements aren't just technical - they may be commercial such as in a Contract. Any constraint has at some stage to be formally captured by a requirement.

More complicated forms are possible by using the date of issue of the standard as a filter. Note that usually, with the exception of national law and safety, once a standard has been applied by contract it will fix the requirements at the issue of the standard applied irrespective of any more recent issues.

MVp-04 Assurance

Version Number

3

Date

8 December 2017

Description

Describes a claim made about any other element with supporting (or opposing) arguments and evidence to establish how and whether a claim is proved or disproved. (as a result of the assessed evidence).

Typical claims for solutions include that a system is safe, fit for purpose and meets its requirements.

Concerns Addressed

Stakeholder	Concern of Stakeholder
Acquirer, Auditor, Builder, Developer, Maintainer, Operator, Owner, User	What are the claims made?
	What is the basis of the claim?
	Is the claim supported by evidence?
Note: stakeholders / roles may be associated with the Enterprise, Concept, Solution, Project, Architecture Task e.g. Builder of Enterprise, Auditor of Solution, Owner of Concept	

Table 2-25 MVp-04 Stakeholder Concerns

Note: stakeholders / roles may be associated with the Enterprise, Concept, Solution, Project, Architecture Task e.g. Builder of Enterprise, Auditor of Solution, Owner of Concept

Anti - Concerns

-

Declared Tuples

Identification of Claim / Forming Argument

- Claim *about* Architecture Description Element
- Claim *has part* Claim
- Claim *supports* Claim
- Claim *opposes* Claim (a counter-claim)
- Argument *supports* Claim
- Argument *opposes* Claim
- Argument *opposes* Argument (a counter-argument)
- Argument *has part* Argument
- Architecture Description Element *traces to* Argument [where architecture of system of interest forms basis of Argument]
- Organisation *makes* Claim
- Role *makes* Claim

Verification of Claim / Argument

As identification +

- Evidence *proves* Claim
- Evidence *disproves* Claim
- Evidence *supports* Argument
- Evidence *opposes* Argument
- Evidence *has part* Evidence

Optional Tuples

Context – Roles

- Job *plays* Role
- Organisation *plays* Role
    - where a typical assurance-related role might be 'Assessor', 'Auditor', 'Design Authority', 'Regulator' in conjunction with
    - Role *extends to* Resource (the object of the claim)

Universal

- Claim *about*, Concern *about*, *traces to* Argument, *traces to* Document, Requirement *governs*, *satisfies* Requirement, Standard *governs*, *satisfies* Standard, Contract *governs*, *satisfies* Contract, *traces to* Contract, *traces to* Requirement, *traces to* Standard may be added to any Architecture Description Element

If any of these optional metamodel elements are added then the appropriate TRAK Master Architecture View must be provided.

Well-Formedness

A MV-04 shall contain:

- at least one Claim (the subject of the view)
- every Claim is connected to the object of the claim (not itself) (Claim *about ...*) or another Claim (Claim *has part* Claim)
- at least one Argument is connected to at least one Claim (Argument *supports* / *opposes* Claim)
- every Argument is connected to a Claim (Claim *supports* / *opposes* Argument) or another Argument (Argument *has part* Argument)
- at least one Evidence is connected to:
    - at least one Argument (Evidence *supports* / *opposes* Argument)
    - and only then that same Evidence may be connected to a Claim (Evidence *proves* / *disproves* Claim)
- every Evidence is directly connected to an Argument (Evidence *supports* / *opposes* Argument) or indirectly connected to an Argument as part of another Evidence (Evidence *has part* Evidence)

Presentation Methods

Design verification of a requirement - the claim is that the design meets the requirement.

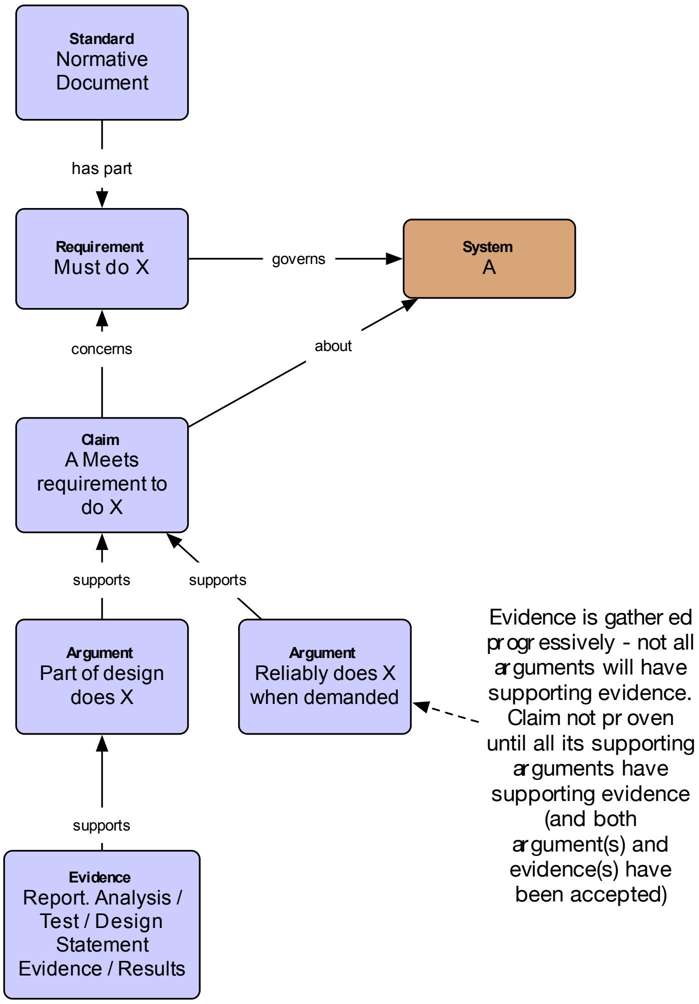

Standard
Normative Document
has part
Requirement
Must do X
governs
System
A
concerns
about
Claim
A Meets requirement to do X
supports
supports
Argument
Part of design does X
Argument
Reliably does X when demanded
Evidence is gather ed progr essively - not all ar guments will have supporting evidence. Claim not pr oven until all its supporting ar guments have supporting evidence (and both ar gument(s) and evidence(s) have been accepted)
supports
Evidence
Report. Analysis / Test / Design Statement Evidence / Results

Figure 2-6-MVp-04 Example 1 - Design Verification

The following shows part of a description from a System Design Authority of a claim about their overall system. This results in a requirement being placed onto the supplier of System C who therefore has to provide a description of their System C meets two requirements placed on the supplied via a System Requirement Requirement Document.

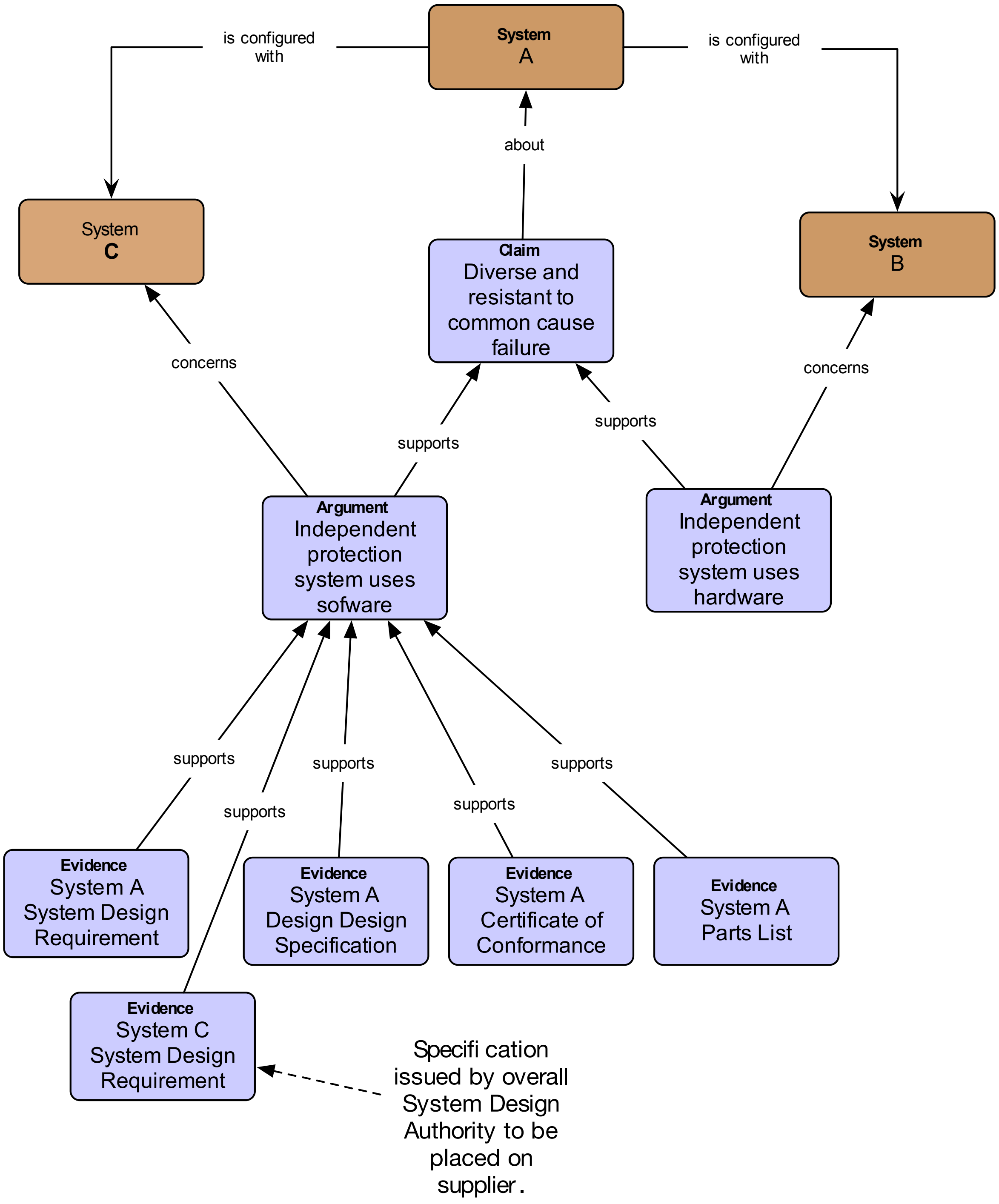

System
A
is configured with
is configured with
about
System
C
Claim
Diverse and resistant to common cause failure
System
B
concerns
concerns
supports
supports
Argument
Independent protection system uses sofware
Argument
Independent protection system uses hardware
supports
supports
supports
supports
supports
Evidence
System A System Design Requirement
Evidence
System A Design Design Specification
Evidence
System A Certificate of Conformance
Evidence
System A Parts List
Evidence
System C System Design Requirement
Specifi cation issued by overall System Design Authority to be placed on supplier .

Figure 2-7-MVp-04 Example 2 - System Design Authority Claim

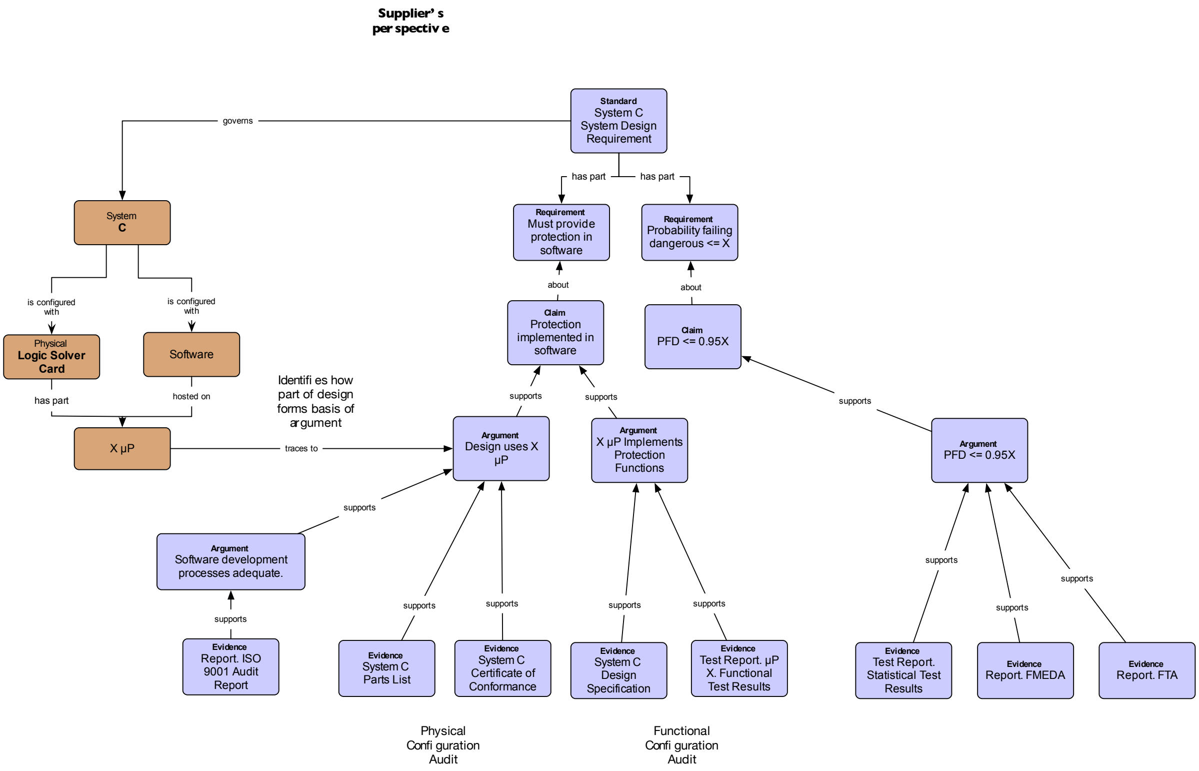


Figure 2-8-MVp-04 Example 3 - Supplier Claim

Views Needed In Order to Construct

Since the object of the Claim (Claim *about* [any TRAK metamodel element]) has to be shown the Master Architecture View for that object will be needed before the MV-04 can be drawn.

The Master Architecture View for any TRAK metamodel element is defined in Table 3-1.

Consistency Rules

- Order of Closing the evidential loop: The assertion that some evidence proves or disproves a claim can only be made after a supporting or opposing argument has been made in respect of this evidence (for the claim):
    - {Evidence *supports* / *opposes* Argument *supports* / *opposes* Argument Claim} **BEFORE** {(same) Evidence *proves* / *disproves* (same) Claim}
- the argument and evidence assertions must be consistent with the assertion that the claim is proved or disproved. For '*proves*' this requires a wholly supporting chain from evidence through the argument(s) to the supported claim(s). For '*disproves*' this requires at least one '*opposes*' relationship (either or both of the evidence doesn't support the argument or the argument doesn't support the claim):
    - **IF** {Evidence *supports* Claim} **THEN** {(same) Evidence *supports* Argument *supports* (same) Claim}

OR

- **IF {**Evidence *opposes* Claim**} THEN EITHER {**(same) Evidence *opposes* Argument *supports* (same) Claim**} OR {**(same) Evidence *supports* Argument *opposes* (same) Claim**}**

Comments

The MV-04 is the master architecture view for Claim, Argument and Evidence.

The supporting (opposing) parts of an Argument or Evidence by inference also support the Claim or Argument respectively to which the top-most 'whole' Argument or Evidence is connected. The part Arguments or part Evidences may also support other Argument or Evidence elements.

A counter-claim is established using 'Claim *opposes* Claim'.

A counter-argument is established using 'Argument *opposes* Argument' (and is usually followed by (same) Argument *opposes* Claim).

When the 'acceptance date' attribute of a Claim, Argument or Evidence element is non-null and valid (not in the future) that element is deemed to have been accepted by the assessor of the claim.

Claims can be made against any element in any TRAK perspective. Claims can be made against a concept, the enterprise and its capabilities and goals, against a system and its ability to realise these capabilities. Claims can also be made against a project, its structure or activities (and thence against the introduction or removal from service of a system). Claims can also be made against standards or against a contract and its requirements.

Applied to the solution perspective this viewpoint supports the creation of a structured safety argument ("safety case"). It can also be used for design verification against the requirements for the design where there is a set of claims that the design meets these requirements. In this sense it could be used to describe how the organisation's processes meet a set of external normative requirements ('Standards' in TRAK).

3 MINIMUM ALLOWED TRAK ARCHITECTURE VIEW SETS

A TRAK Master Architecture View is a view type within an architecture description or model which acts as the master source or record for one or more TRAK metamodel elements. For example, all Resource (System, Physical, Software, Organisation, Job and Role) objects have to be first declared on one or more SV-01 Solution Structure architecture views. In other words the SV-01 is the Master Architecture View for Resource. This stops folks from introducing object types and relationships in views where the focus is inappropriate. In this particular case the focus of the SV-01 is on structure and therefore you wouldn't want structural relationships to be first introduced into the SV-02 where the focus is on interactions between the resources.

TRAK Architecture Views deliberately overlap for reasons of readability, context and navigability. Since views contain a mixture of metamodel elements there is therefore a dependency between views. For example, the SV-02 Solution Resource Interaction view requires a minimum Resource and Resource Interaction. This in turn requires Resource to first be declared on the SV-01. If you need to produce a SV-02 the minimum allowed view set is at least a SV-01 and SV-02, the MV-01 Architecture Description Dictionary that defines each elementand the MV-02 Architecture Description Design Record that describes the task in this case.

The dependency between TRAK architecture views and the master architecture views is shown in Figure 3-1, Figure 3-2 ,~~and~~ Figure 3-3 and Figure 3-4.

The minimum allowed view sets within a TRAK-compliant architecture description are defined in Table 3-2. The dependencies between view sets are shown in Figure 3-5.

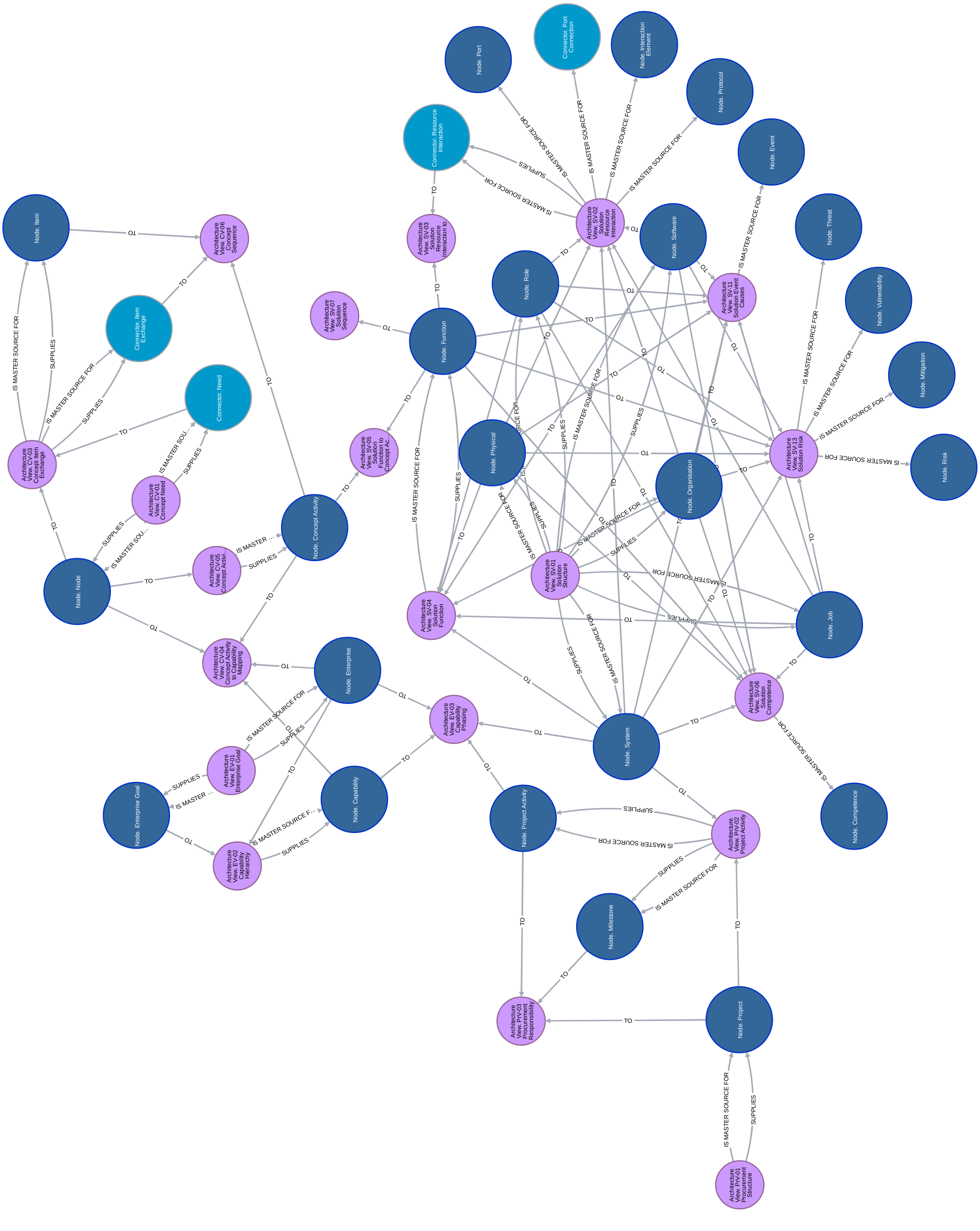

Node. Port
Connector. Port Connection
Node. Interaction Element
Node. Protocol
Node. Event
Connector. Resource Interaction
Architecture View. SV-03 Solution Resource Interaction to
Architecture View. SV-02 Solution Resource Interaction
Node. Software
Architecture View. SV-11 Solution Event Causes
Node. Threat
Node. Vulnerability
Node. Mitigation
Node. Risk
Architecture View. SV-13 Solution Risk
Node. Role
Node. Function
Architecture View. SV-07 Solution Sequence
Architecture View. SV-05 Solution Function to Concept Ac...
Node. Physical
Node. Organisation
Architecture View. SV-01 Solution Structure
Node. Job
Architecture View. SV-04 Solution Function
Architecture View. SV-06 Solution Competence
Node. Competence
Node. System
Architecture View. EV-03 Capability Phasing
Node. Enterprise
Node. Capability
Architecture View. EV-01 Enterprise Goal
Node. Enterprise Goal
Architecture View. EV-02 Capability Hierarchy
Node. Project Activity
Architecture View. PrV-02 Project Activity
Node. Milestone
Architecture View. PrV-03 Procurement Responsibility
Node. Project
Architecture View. PrV-01 Procurement Structure
Node. Item
Architecture View. CV-06 Concept Sequence
Connector. Item Exchange
Connector. Need
Architecture View. CV-03 Concept Item Exchange
Architecture View. CV-01 Concept Need
Node. Node
Architecture View. CV-05 Concept Activi...
Node. Concept Activity
Architecture View. CV-04 Concept Activity to Capability Mapping
TO
SUPPLIES
IS MASTER SOURCE FOR

Figure 3-1 Master Architecture Views Create View Dependencies

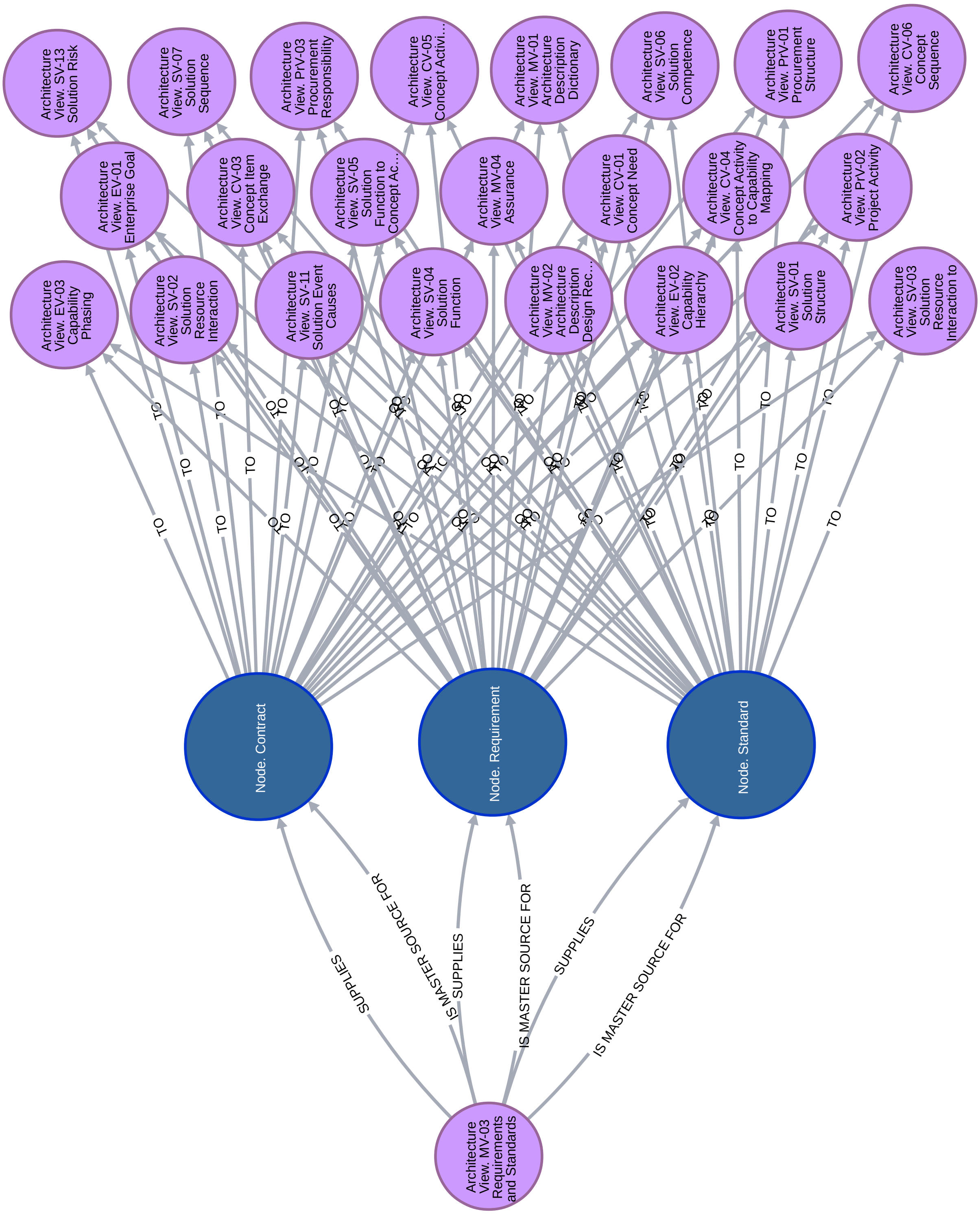

Architecture View. SV-13 Solution Risk
Architecture View. SV-07 Solution Sequence
Architecture View. PrV-03 Procurement Responsibility
Architecture View. CV-05 Concept Activi...
Architecture View. MV-01 Architecture Description Dictionary
Architecture View. SV-06 Solution Competence
Architecture View. PrV-01 Procurement Structure
Architecture View. CV-06 Concept Sequence
Architecture View. EV-01 Enterprise Goal
Architecture View. CV-03 Concept Item Exchange
Architecture View. SV-05 Solution Function to Concept Ac...
Architecture View. MV-04 Assurance
Architecture View. CV-01 Concept Need
Architecture View. CV-04 Concept Activity to Capability Mapping
Architecture View. PrV-02 Project Activity
Architecture View. EV-03 Capability Phasing
Architecture View. SV-02 Solution Resource Interaction
Architecture View. SV-11 Solution Event Causes
Architecture View. SV-04 Solution Function
Architecture View. MV-02 Architecture Description Design Rec...
Architecture View. EV-02 Capability Hierarchy
Architecture View. SV-01 Solution Structure
Architecture View. SV-03 Solution Resource Interaction to
TO
Node. Contract
Node. Requirement
Node. Standard
SUPPLIES
IS MASTER SOURCE FOR
Architecture View. MV-03 Requirements and Standards

Figure 3-2- Master Architecture Views – Contracts, Requirements & Standards

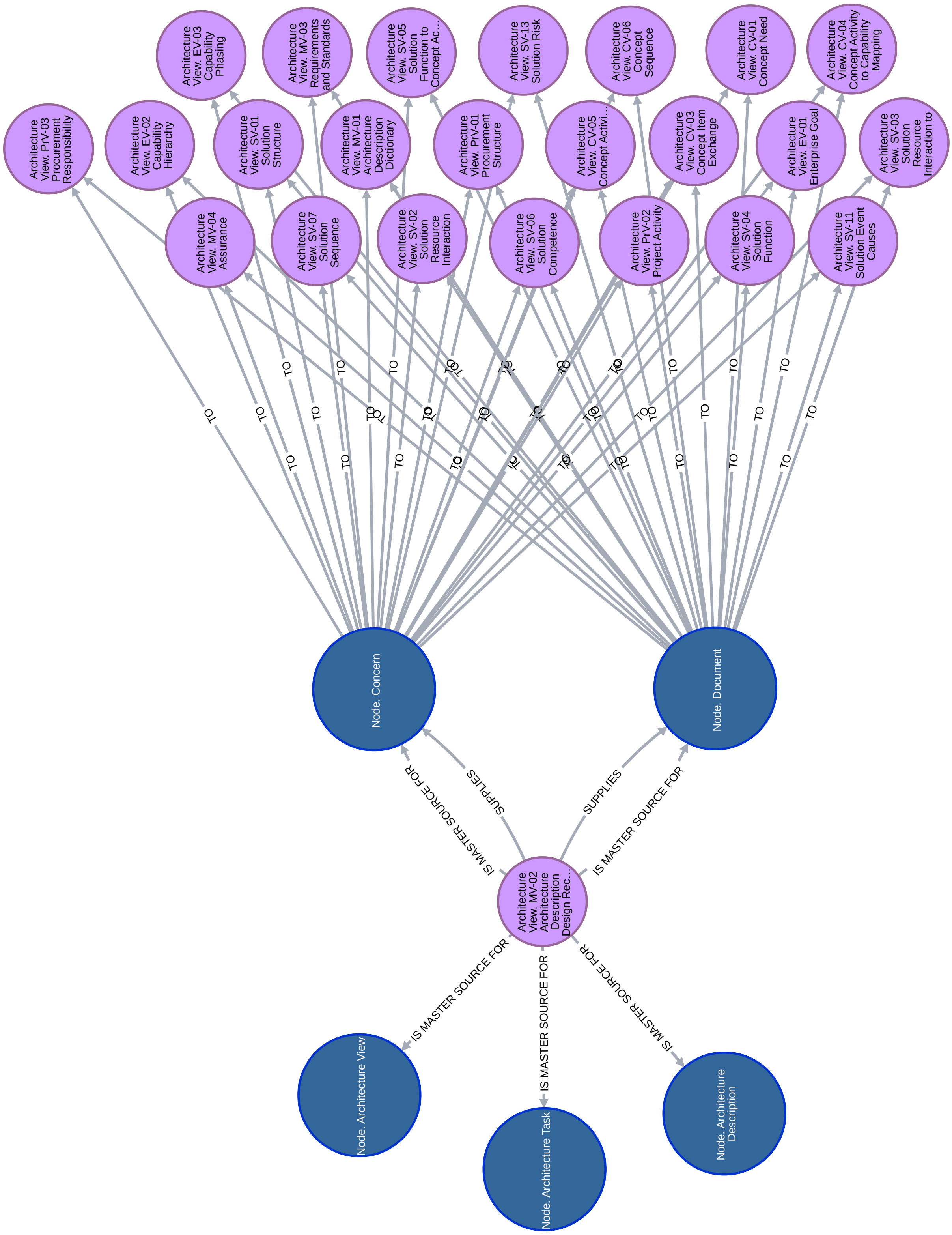

Architecture View. EV-03 Capability Phasing
Architecture View. MV-03 Requirements and Standards
Architecture View. SV-05 Solution Function to Concept Ac...
Architecture View. SV-13 Solution Risk
Architecture View. CV-06 Concept Sequence
Architecture View. CV-01 Concept Need
Architecture View. CV-04 Concept Activity to Capability Mapping
Architecture View. PrV-03 Procurement Responsibility
Architecture View. EV-02 Capability Hierarchy
Architecture View. SV-01 Solution Structure
Architecture View. MV-01 Architecture Description Dictionary
Architecture View. PrV-01 Procurement Structure
Architecture View. CV-05 Concept Activi...
Architecture View. CV-03 Concept Item Exchange
Architecture View. EV-01 Enterprise Goal
Architecture View. SV-03 Solution Resource Interaction to
Architecture View. MV-04 Assurance
Architecture View. SV-07 Solution Sequence
Architecture View. SV-02 Solution Resource Interaction
Architecture View. SV-06 Solution Competence
Architecture View. PrV-02 Project Activity
Architecture View. SV-04 Solution Function
Architecture View. SV-11 Solution Event Causes
TO
Node. Concern
Node. Document
SUPPLIES
IS MASTER SOURCE FOR
SUPPLIES
IS MASTER SOURCE FOR
Architecture View. MV-02 Architecture Description Design Rec...
IS MASTER SOURCE FOR
IS MASTER SOURCE FOR
IS MASTER SOURCE FOR
Node. Architecture View
Node. Architecture Task
Node. Architecture Description

Figure 3-3 - Master Architecture Views – Assurance Elements

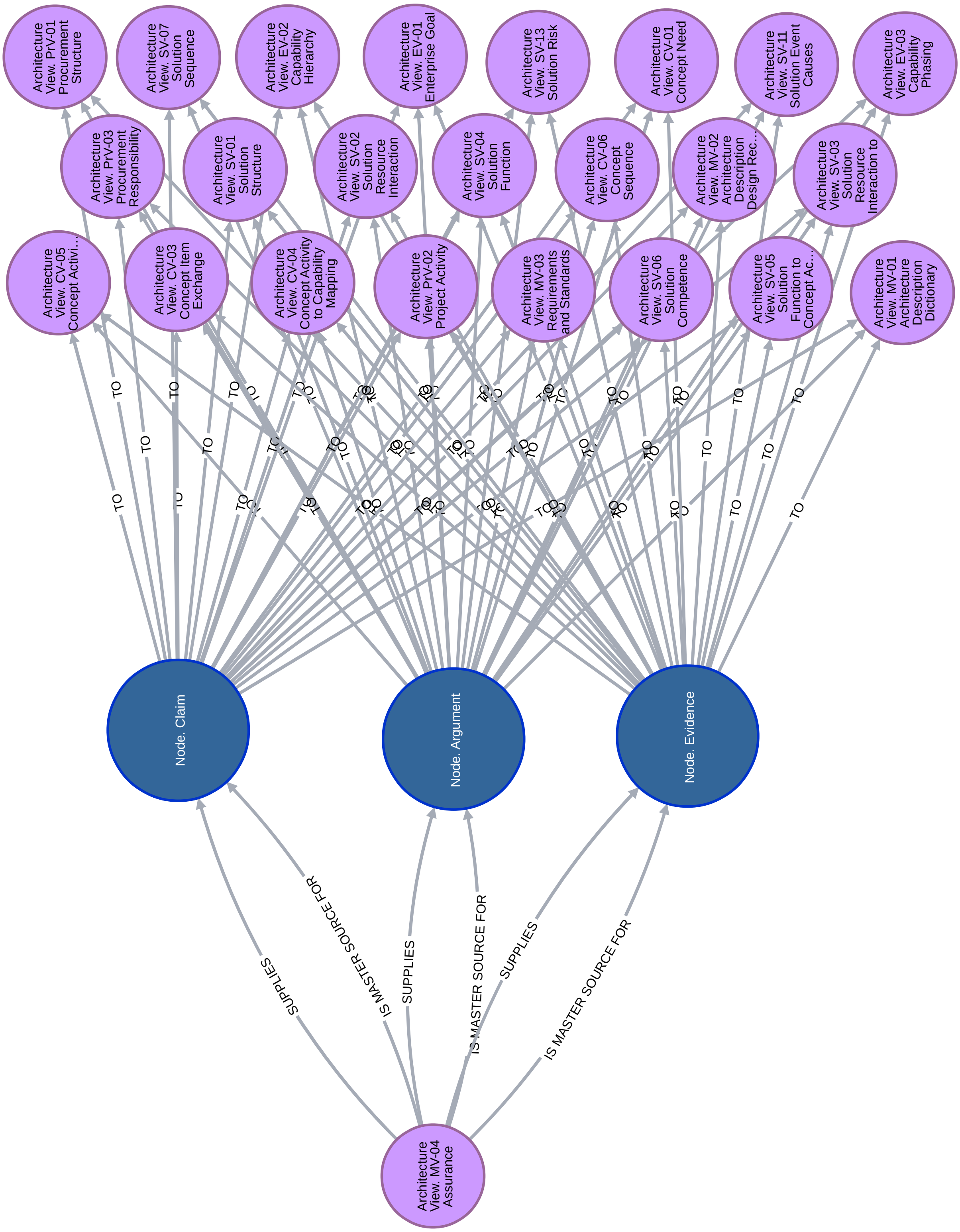

Architecture View. PrV-01 Procurement Structure
Architecture View. SV-07 Solution Sequence
Architecture View. EV-02 Capability Hierarchy
Architecture View. EV-01 Enterprise Goal
Architecture View. SV-13 Solution Risk
Architecture View. CV-01 Concept Need
Architecture View. SV-11 Solution Event Causes
Architecture View. EV-03 Capability Phasing
Architecture View. PrV-03 Procurement Responsibility
Architecture View. SV-01 Solution Structure
Architecture View. SV-02 Solution Resource Interaction
Architecture View. SV-04 Solution Function
Architecture View. CV-06 Concept Sequence
Architecture View. MV-02 Architecture Description Design Rec...
Architecture View. SV-03 Solution Resource Interaction to
Architecture View. CV-05 Concept Activi...
Architecture View. CV-03 Concept Item Exchange
Architecture View. CV-04 Concept Activity to Capability Mapping
Architecture View. PrV-02 Project Activity
Architecture View. MV-03 Requirements and Standards
Architecture View. SV-06 Solution Competence
Architecture View. SV-05 Solution Function to Concept Ac...
Architecture View. MV-01 Architecture Description Dictionary
TO
Node. Claim
Node. Argument
Node. Evidence
SUPPLIES
IS MASTER SOURCE FOR
SUPPLIES
IS MASTER SOURCE FOR
SUPPLIES
IS MASTER SOURCE FOR
Architecture View. MV-04 Assurance

Figure 3-4 - Master Architecture Views – Architecture Task Description Elements

TRAK Metamodel Element	Master Architecture View
Architecture Description	MV-02 Architecture Description Design Record
Architecture Task	MV-02 Architecture Description Design Record
Architecture View	MV-02 Architecture Description Design Record
Argument	MV-04 Assurance
Capability	EV-02 Capability Hierarchy
Claim	MV-04 Assurance
Competence	SV-06 Solution Competence
Concept Activity	CV-05 Concept Activity
Concern	MV-02 Architecture Description Design Record
Contract	MV-03 Requirements & Standards
Document	MV-02 Architecture Description Design Record
Enterprise	EV-01 Enterprise Goals
Enterprise Goal	EV-01 Enterprise Goals
Event	SV-11 Solution Event Causes
Evidence	MV-04 Assurance
Function	SV-04 Solution Function
Interaction Element	SV-02 Solution Resource Interaction
Item	CV-03 Concept Item Exchange
Item Exchange	CV-03 Concept Item Exchange
Job	SV-01 Solution Structure
Metric	The Metric should be declared on the EV-02, CV-05 orSV-04 on which it quantifies the Capability, Concept Activity or Function respectively.
Milestone	PrV-02 Procurement Timeline

TRAK Metamodel Element	Master Architecture View
Mitigation	SV-13 Solution Risk
Need	CV-01 Concept Need
Node	CV-01 Concept Need
Organisation	SV-01 Solution Structure
Physical	SV-01 Solution Structure
Port	SV-02 Solution Resource Interaction
Port Connection	SV-02 Solution Resource Interaction
Project	PrV-01 Project Structure
Project Activity	PrV-02 Procurement Timeline
Protocol	SV-02 Solution Resource Interaction
Requirements	MV-03 Requirements & Standards
Resource Interaction	SV-02 Solution Resource Interaction
Risk	SV-13 Solution Risk
Role	SV-01 Solution Structure
Software	SV-01 Solution Structure
Standard	MV-03 Requirements & Standards
System	SV-01 Solution Structure
Threat	SV-13 Solution Risk
Vulnerability	SV-13 Solution Risk
Note: Only metamodel block elements that can appear on architecture views are listed I.	

Table 3-1 Master Architecture View for Each TRAK Metamodel Element

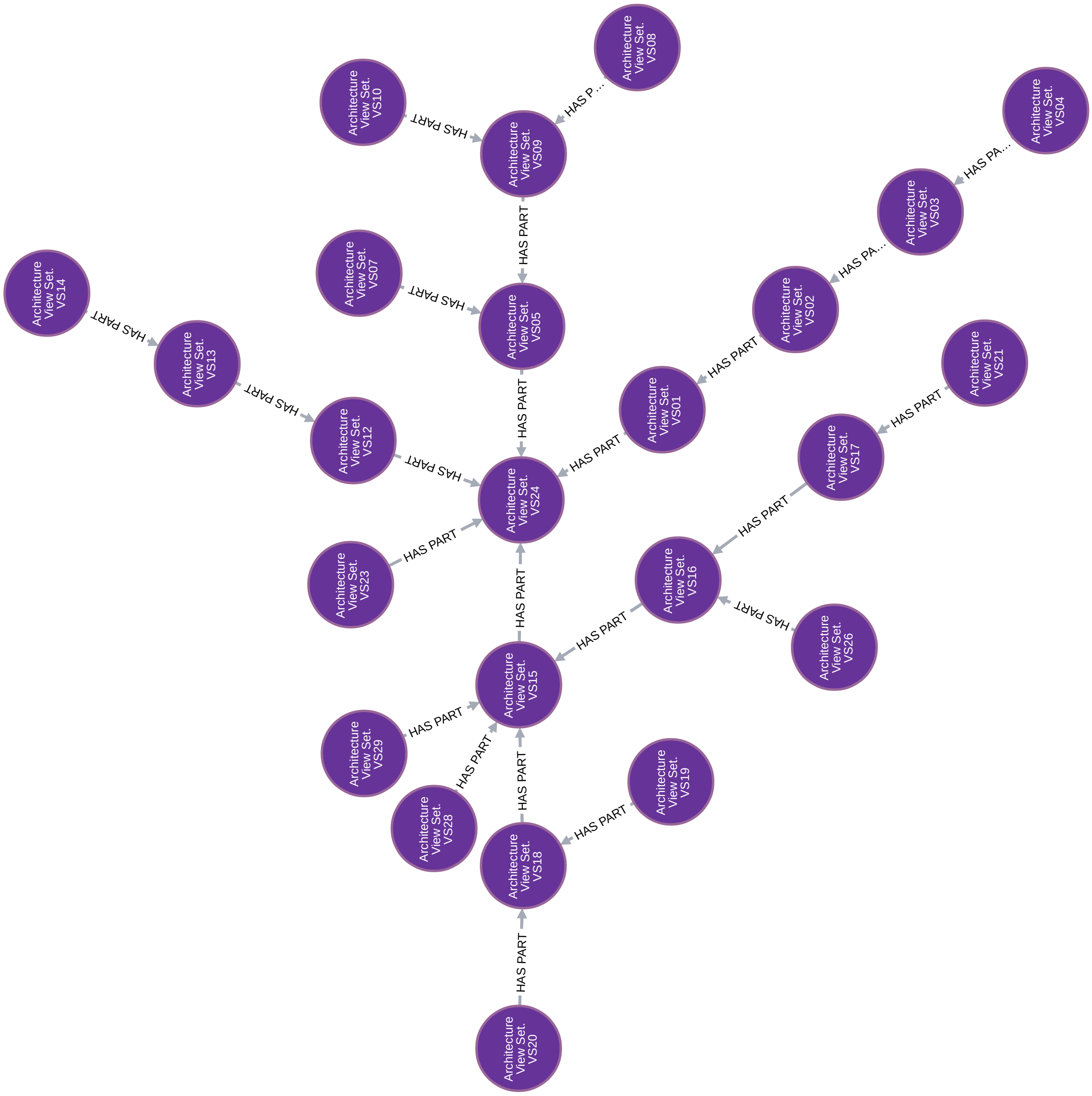

Architecture View Set. VS08
HAS P…
Architecture View Set. VS10
HAS PART
Architecture View Set. VS09
Architecture View Set. VS04
HAS PA…
Architecture View Set. VS03
HAS PA…
HAS PART
Architecture View Set. VS14
HAS PART
Architecture View Set. VS07
HAS PART
Architecture View Set. VS05
Architecture View Set. VS02
HAS PART
Architecture View Set. VS13
HAS PART
Architecture View Set. VS21
Architecture View Set. VS01
HAS PART
HAS PART
Architecture View Set. VS12
HAS PART
HAS PART
Architecture View Set. VS17
Architecture View Set. VS24
HAS PART
HAS PART
Architecture View Set. VS23
Architecture View Set. VS16
HAS PART
HAS PART
HAS PART
Architecture View Set. VS26
Architecture View Set. VS15
HAS PART
Architecture View Set. VS29
HAS PART
HAS PART
Architecture View Set. VS19
Architecture View Set. VS28
HAS PART
Architecture View Set. VS18
HAS PART
Architecture View Set. VS20

Figure 3-5 Architecture View Set Dependencies (Graph)

ID	Date	View	View Set Needed	Justification
VS1	29/04/2010	EV-01	MV-01 + MV-02 + EV-01	EV-01 is master view for Enterprise, Enterprise Goal
VS2	29/04/2010	EV-02	MV-01 + MV-02 + EV-02 + EV-01	EV-01 is master view for Enterprise, EV-02 for Capability
VS3	29/04/2010	EV-03	MV-01 + MV-02 + EV-03 + EV-01 + EV-02	EV-03 is a mapping view. For planned capability phasing Enterprise provides time duration. EV-01 is master view for Enterprise, EV-02 is master view for Capability
VS4	29/04/2010	EV-03	MV-01 + MV-02 + EV-03 + EV-01 + EV-02 + PrV-02 + PrV-01 + SV-01	EV-03 is a mapping view. For realised capability phasing vs planned. PrV-02 provides actual time duration. SV-01 is master view for Resource, PrV-01 is master view for Project, EV-02 is master view for Capability, EV-01 is master view for Enterprise.
VS5	29/04/2010	CV-01	MV-01 + MV-02 + CV-01	CV-01 is master view for Node, Need
VS6	29/04/2010	CV-02		[CV-02 deleted]
VS7	29/04/2010	CV-03	MV-01 + MV-02 + CV-03 + CV-01	CV-03 is master view for Item, Item Exchange. CV-01 is master view for Node, Need
VS8	29/04/2010	CV-04	MV-01 + MV-02 + CV-04 + CV-05 + CV-01 + EV-02 + EV-01	CV-04 is a mapping view. CV-05 is master view for Concept Activity, CV-01 is master view for Node, EV-02 is master view for Capability, EV-01 is master view for Enterprise
VS9	29/04/2010	CV-05	MV-01 + MV-02 + CV-05 + CV-01	CV-05 is master view for Concept Activity, CV-01 is master view for Node

ID	Date	View	View Set Needed	Justification
VS10	29/04/2010	CV-06	MV-01 + MV-02 + CV-06 + CV-01 + CV-05 + CV-03	CV-01 is master view for Node. CV-05 is master view for Concept Activity. CV-03 is master view for Item, Item Exchange
VS11	01/04/2010	OV-07		Subsumed into MV-03. See VS25
VS12	15/02/2010	PrV-01	MV-01 + MV-02 + PrV-01	PrV-01 is master view for Project
VS13	15/02/2010	PrV-02	MV-01 + MV-02 + PrV-02 + PrV-01	PrV-02 is master view for Milestone, Project Activity. PrV-01 is master view for Project
VS14	15/02/2010	PrV-03	MV-01 + MV-02 + PrV-03 + PrV-02 + PrV-01 + SV-01	PrV-02 is master for Milestone. PrV-01 is master view for Project. SV-01 is master view for Resource including Role for Role 'extends to' Resource relationship.
VS15	15/02/2010	SV-01	MV-01 + MV-02 + SV-01	SV-01 is master view for Resource (System, Physical, Software, Organisation, Job, Role)
VS16	15/02/2010	SV-02	MV-01 + MV-02 + SV-02 + SV-01	SV-02 is master view for Resource Interaction, Port, Port Connection, Protocol, Interaction Element. SV-01 is master view for Resource.
VS17		SV-03	MV-01 + MV-02 + SV-03 + SV-02 + SV-01 + SV-04	SV-03 is a mapping view. SV-02 is master view for Resource Interaction, Port, Port Connection, Protocol, Interaction Element. SV-01 is master view for Resource. SV-04 is master view for Function.
VS18		SV-04	MV-01 + MV-02 + SV-04 + SV-01	SV-04 is master view for Function. SV-01 is master view for Resource.

ID	Date	View	View Set Needed	Justification
VS19	29/04/10	SV-05	MV-01 + MV-02 + SV-05 + SV-04 + SV-01 + CV-05 + CV-01	SV-05 is a mapping view. SV-04 is master view for Function. SV-01 is master view for Resource. CV-05 is master view for Concept Activity. CV-01 is master view for Node.
VS20		SV-06	MV-01 + MV-02 + SV-06 + SV-04 + SV-01	SV-06 is master view for Competence. SV-04 is master view for Function. SV-01 is master view for Resource.
VS21		SV-07 (functional form)	MV-01 + MV-02 + SV-07 + SV-04 + SV-03 + SV-02 + SV-01	SV-04 is master view for Function. SV-02 is master view for Resource Interaction, Interaction Element. SV-01 is master view for Resource. SV-03 created as result by identifying exchange of Interaction Element in combination with associated Function and Resource Interaction
VS26	08/03/2011	SV-07 (Resource Interactio form)	MV-01 + MV-02 + SV-07 + SV-03 + SV-02 + SV-01	SV-02 is master view for Resource Interaction, Interaction Element. SV-01 is master view for Resource. SV-03 created as result by identifying exchange of Interaction Element in combination with associated Function and Resource Interaction
VS22	01/04/2010	SV-10		Subsumed into MV-03. See VS25
VS28	01/01/2016	SV-11	MV-01 + MV-02 + SV-11 + SV-01	SV-11 is master view for Event. SV-01 is master view for Resource.

ID	Date	View	View Set Needed	Justification
VS29	01/01/2016	SV-13	MV-01 + MV-02 +SV-13 + SV-01	SV-13 is master view for Mitigation, Vulnerability, Risk and Threat.
VS23		MV-01	MV-01 + MV-02 + any other views	MV-01 provides glossary for architecture description elements used in all other views.
VS24		MV-02	MV-01 + MV-02 + any other views	MV-02 is master view for Architecture Task, Concern, If Concern appears in any other view then MV-02 must be produced. A MV-02 must be produced to describe/document any AD.
VS25	01/04/2010	MV-03	MV-01 + MV-02 + MV-03 + any other views	MV-03 is master view for Requirement, Standard. If Requirement or Standard appears in any other view then MV-03 must be produced.
VS27	30/12/2014	MV-04	MV-01 + MV-02 + MV-04 + claim subject element master view	MV-04 is master view for Claim, Argument and Evidence. Requires master view for element that is the subject of the claim.
Note: Every architecture tuple within a AD must appear on a view i.e. an AD must contain architecture and must have architecture views (BLV-4.1)				

Table 3-2 Minimum Allowed View Sets

4 ORIGINAL TRAK BASELINE VS MODAF 1.2

TRAK Architecture Views

For every TRAK architecture view at initial release, the following table makes a brief comparison against the view(s) in MODAF® 1.2.003 [Ref. 3]

Post Release Note:

Since the initial release of TRAK there have been some name changes:-

- ‘Capability Perspective’ is now ‘Enterprise Perspective’
- ‘Operational Perspective’ is now ‘Concept Perspective’

which affects TRAK viewpoint names. The current list of TRAK viewpoints is defined in Table 2-1.

TRAK View(point)	Closest MODAF® View(s)	TRAK Differences
Capability Perspective	Strategic Viewpoint	
CV-01 Enterprise Goal	StV-1 Enterprise Vision	Must be produced if a Enterprise or Enterprise Goal is present in model.
CV-02 Capability Hierarchy	StV-2 Capability Taxonomy	Must be produced if a Capability is present in model.
CV-03 Capability Phasing	StV-3 Capability Phasing	
Operational Perspective	Operational Viewpoint	
OV-01 Operational Topology	OV-2 Operational Node Relationship Description	Must be produced if a Node or Need is present in model.
OV-02 Operational Concept	OV-1a,b,c High Level Operational Concept Graphic / Operational Concept Description / Operational Performance Attributes	
OV-03 Operational Item Exchange	OV-3 Operational Information Exchange Matrix	Must be produced if an Item or Item Exchange is present in model.
OV-04 Operational Activity to Capability Mapping	StV-6 Operational Activity to Capability Mapping	
OV-05 Operational Activity	OV-5 Operational Activity Model	Must be produced if an Operational Activity is present in model.

TRAK View(point)	Closest MODAF® View(s)	TRAK Differences
OV-06 Operational Sequence	OV-6 b,c / State Transition Description / Event-Trace Description	
Procurement Perspective	Acquisition Viewpoint	
PrV-01 Procurement Structure	AcV-1 Acquisition Clusters	Must be produced if a Project is present in model.
PrV-02 Procurement Timeline	AcV-2 Programme Timelines	Must be produced if a Project Activity or Milestone is present in model.
PrV-03 Procurement Responsibility	- none -	Shows extent/jurisdiction of responsibility at specified time.
Solution Perspective	Systems Viewpoint	
SV-01 Solution Structure	SV-1 Resource Interaction Description	TRAK::SV-01 only includes structural relationships not interactions. Organisational governance of MODAF::OV-4 shown in TRAK::SV-01. Must be produced if a Resource (System, Physical, Software, Organisation, Job, Role) is present in model.
SV-02 Solution Resource Interaction	SV-1 Resource Interaction Specification / SV-2a,b System Port Specification, System Port Connectivity Description / OV-04	Must be produced if a Resource Interaction, Port, Port Connection, Protocol or Interaction Element is present in model. Note that in TRAK any Resource can have a Port and exchange an Interaction Element.
SV-03 Solution Resource Interaction to Function Mapping	- no equivalent -	To ensure every interaction has a functional justification.
SV-04 Solution Functionality	SV-4 Functionality Description	Must be produced if a Function is present in model.
SV-05 Solution Function to Operational Activity Mapping	SV-5 Function to Operational Activity / Service Function Traceability Matrix	
SV-06 Solution Competence	- no equivalent -	For Human Factors, Human Resources use. Must be produced if a Competence is present in model.

TRAK View(point)	Closest MODAF® View(s)	TRAK Differences
SV-07 Solution Sequence	SV-10b Resource State Transition Description	
Management Perspective	All Views Viewpoint	
MV-01 Architecture Description Dictionary	AV-2 Integrated Dictionary	
MV-02 Architecture Description Design Record	AV-1 Overview and Summary Information	As a minimum takes ISO 42010 topics. In TRAK includes AD, View and used to describe architectural task in a view (not just text). Must be produced.
	Technical Viewpoint	
MV-03 Requirements & Standards	SV-10a Resource Constraints Specification OV-6a Operational Rules Model TV-1 Standards Profile, TV-2 Standards Forecast Although none identify effect of a contract.	Must be produced if a solution Requirement is present in model. TRAK::MV-03 also used to show how contract applies standards at issue. TRAK::Standard metamodel element has more dependency relationships with itself - precedence. Must be produced if a Standard is present in model.

Table 4-1 - TRAK Views vs MODAF® 1.2 Views

MODAF® 1.2 Views Not Present in TRAK

The following MODAF® 1.2 [Ref. 3] views are not present in TRAK at initial release.

MODAF® 1.2 View	Comment
Strategic Views Viewpoint	
StV-4 Capability Dependencies	Must be produced if a Enterprise or Enterprise Goal is present in model.
StV-5 Capability to Organisation Deployment Mapping v1.2	Must be produced if a Capability is present in model.
StV-6 Operational Activity to Capability Mapping	View moved into Operational Perspective as TRAK:: OV-04 so that mapping consistently upwards from Function - Operational Activity - Capability.
Operational Views Viewpoint	
OV-4 Organisational Relationships Chart	MODAF::OV-4 only contains Resource i.e. solution and shows structure and interactions hence mapped into TRAK::SV-01 and TRAK::SV-02.
OV-6a Operational Rules Model	All rules, constraints and forms of requirement are covered by TRAK::MV-03
OV-7 Information Model	
System Views Viewpoint	
SV-7 Resource Performance Parameters Matrix	Parameters usually associated with a requirement and often function - covered via TRAK::SV-04 and TRAK::SV-10
SV-8 Capability Configuration Management	
SV-9 Technology & Skills Forecast	Standards are covered by the TRAK::MV-03 and skills by the TRAK::SV-06
SV-10a Resource Constraints Specification	All rules, constraints and forms of requirement are covered by TRAK::MV-03
SV-11 Physical Schema	
SV-12 Service Provision	Any requirement can be expressed on a TRAK::MV-03
Service-Oriented Views Viewpoint	
SOV-1 Service Taxonomy	A service, if needed, could be represented using a combination Software, Port, Port Connection, Requirement and Function in the solution perspective. In short it would be treated a System comprising only Software. If an implementation-free description of something is needed the operational perspective views can be used.
SOV-2 Services Interface Specification	
SOV-3 Capability to Service Mapping	
SOV-4a Service Constraints	
SOV-4b Service State Model	
SOV-4c Service Interaction Specification	
SOV-5 Service Functionality	
Technical Standards Views Viewpoint	

MODAF® 1.2 View	Comment
TV-1 Standards Profile / TV-2 Standards Forecast	Merged into single TRAK::MV-03

Table 4-2 - MODAF® Views Not Present in TRAK at Initial Release

REFERENCES

Ref. 1 ISO/IEC/IEEE 42010:2011 Systems & Software Engineering - Architecture Description

Ref. 2 DODAF 1.5[3]. The US Department of Defense Architecture Framework.

Ref. 3 MODAF 1.2. The UK Ministry of Defence Architecture Framework. www.mod.uk/modaf

Ref. 4 TRAK00004 TRAK Architecture Framework. sf.net/p/trak

Ref. 5 TRAK00002 TRAK Architecture Metamodel. sf.net/p/trakmetamodel

Ref. 6 Sourceforge. Known implementations of TRAK. trak.sourceforge.net/implementations.html

Ref. 7 TRAK UML profile. sf.net/p/trakumlprofile

Ref. 8 MDG Technology for TRAK. sf.net/p/mdgfortrak

Ref. 9 TRAK Viewpoints project RSS Feed - files
http://sourceforge.net/api/file/index/project-id/304405/mtime/desc/limit/20/rss

Ref. 10 TRAK Viewpoints project RSS Feed - feature requests http://sourceforge.net/p/trakviewpoints/feature-requests-viewpoints/feed.rss

Ref. 11 TRAK Viewpoints project RSS Feed - bugs http://sourceforge.net/p/trakviewpoints/bugs-viewpoints/feed.rss

Ref. 12 TRAK Viewpoints project RSS Feed - news - http://sourceforge.net/p/trakviewpoints/news/feed

Ref. 13 TRAK00005 TRAK. Implementation. Architecture Description Elements.
http://sourceforge.net/projects/trak/files/Implement%20TRAK/

Ref. 14 GNU Free Documentation License 1.3. http://www.gnu.org/licenses/fdl-1.3.html

[3] The current version of DODAF is 2.0 at http://cio-nii.defense.gov/sites/dodaf20/

INDEX

BACK COVER