# Modelling Boundaries

## Expressing Intent, Capability, Partial Visibility etc.

Nigel Davis (Ciena)

20221102

Presented at IETF 115 London (nmrg (Monday 7 Nov, 9:30 – 11:30) and netmod (Tuesday 8 Nov, 9:30 – 11:30))

Related material at https://datatracker.ietf.org/doc/html/draft-davis-netmod-modelling-boundaries-00

IETF®

# Overview

- Observations
- Essential Problem/Solution
- Advanced solution concepts
- Sketch of progressive narrowing
- System Assembly and Occurrence
- Partial compatibility consideration
- Target and next steps

Notes:

- Content: A brief sketch of draft-davis-netmod-modelling-boundaries
  - There is not time in this presentation to dig into each area
- Terminology: Each key term used in this document has specific local meaning
  - It is probable that the definitions here are currently too vague to ensure full shared understanding
- Background: 24+ years of grappling with this challenge

# Observations

- Advanced control solutions require modelling/representation of:
  - Intent, Capability, Partial Visibility, Policy, Negotiation etc.
    - Each requires extensive modelling of constraints, focusses and of uncertain/vague/interrelated boundaries
- Representations develop through recursive/progressive narrowing
  - No specific narrowing number/levels
  - Stop when right level for viewpoint
- Narrowing results in an Occurrence definition
  - An Occurrence at one narrowing level is narrowed to an Occurrence at the next level
  - A traditional “instance” is an Occurrence with very tight narrowing
- Different Occurrences can be assembled to form a structure
  - A collection of functional Occurrences assemble into a system
    - Where an Occurrences is essentially a component
  - In a system there maybe many similar Occurrences
    - Each with a different role or in a different position etc

# Essential Problem/Solution

- Modelling approaches and processes need fundamental support for
  - Recursive narrowing/tightening of constraints from one representation to the next
  - Any modelled thing at any level to be treated consistently as an Occurrence
  - Each property at any level of narrowing to be defined in terms of constraints
  - A single expression method for a mix of ranges, focusses and single value statements
- Conventional modeling techniques do not naturally support the above
  - Meta-Class, Class and Instance are distinct and different in representation
  - Property types emphasize single value statements
- YANG is used to define properties/structure
  - YANG appears to be appropriately formed to accommodate the approach
  - Enhancement to be within, as extensions to, and compatible with current definitions

# Advanced solution concepts

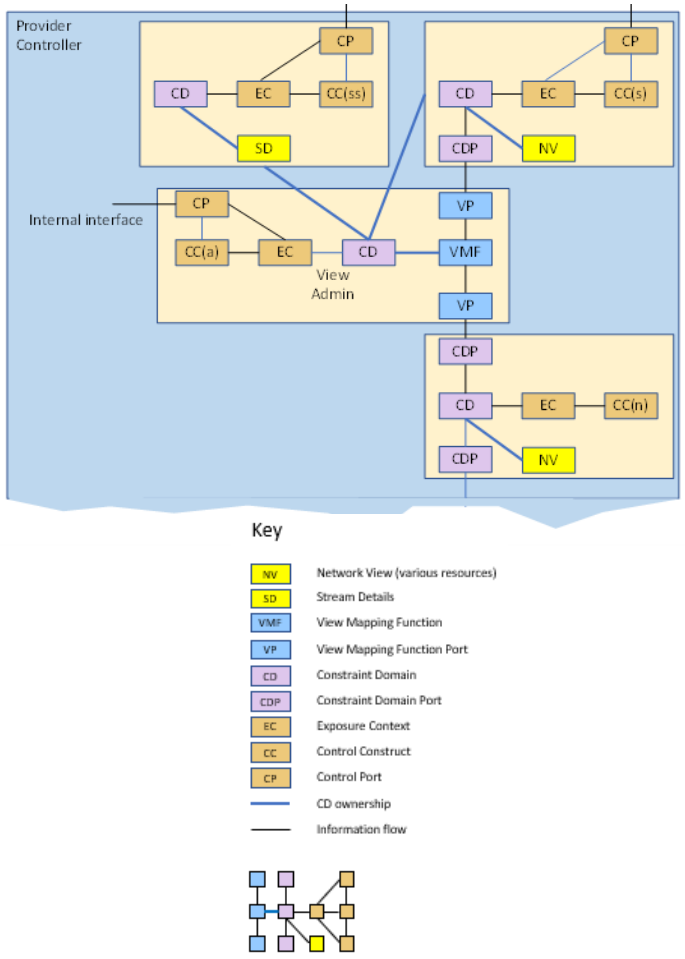
- Specification of Intent (Expectation/Intention used in the I-D)
  - Statement of desired outcome in terms of constraints
  - Includes statement of preference and acceptable value ranges etc.
- Specification of Capability
  - Statement of opportunity for behaviour to be exhibited (the effect, not the realization)
  - Includes statement of possible ranges (constraints) and interdependencies
- Expression of Partial Visibility of state etc.
  - Statement made in a noisy/lossy/imprecise environment about behaviour/characteristics
  - Includes statements of probability, uncertainty and vagueness

# Sketch of progressive narrowing

| Bit rate example illustration | Type | Constraint | Comment |
|---|---|---|---|
| 1) Standard | Integer | Boundless (Hz) | All encompassing range |
| 2a) Vendor solution A | Integer | 10G-100G | High end technology constraints |
| 2b) Vendor solution B | Integer | 5G-50G | Technology price point constraint |
| 3a) Application A | Integer | 10G-50G | Deployment combinatorial limit |
| 3b) Application A at point x | Integer | 30G-50G | A higher capacity point in the application |
| 3c) Application A point x busy hour | Integer | 50G | Busy hour setting |
| 4a) PoC (Proof of Concept) | Enumeration | 30G, 40G, 50G | A quantized selection for a PoC |
| 4b) PoC Use case | Enumeration | 40G | A specific use case with a fixed bit rate |
| 5a) Intent instance X | Enumeration | 30G, 40G | The lowest level view still enables range statements as there is delegated control |
| 5b) Intent instance y | Enumeration | 40G | A specific tight constraint on the delegated control OR a single value for an “instance” |

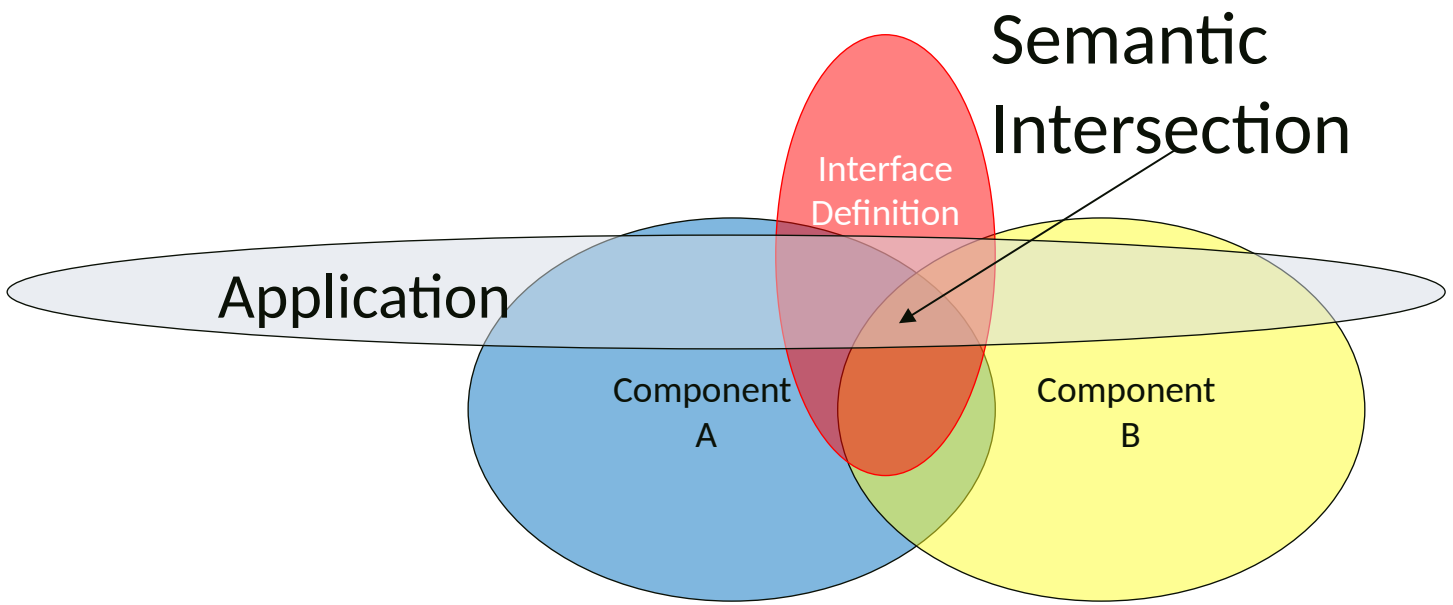
# System Assembly and Occurrence

- An Occurrence at one level of specification is a narrow use of an Occurrence at the previous higher level of specification
- There will be many Occurrences at a lower level derived from an Occurrence at a higher level.
- The Occurrences at the lower level will be distinct from each other.
- A System structure may make repeated use of same type of component, i.e., of Occurrences derived from the same previous higher level Occurrence
- An Occurrence is a use of a particular component type in a system structure where each use may have subtly different narrowing of capabilities to each other use
- The term Occurrence appears in ONF work (see TR-512_v1.5_OnfCoreIm-info.zip)
- Capability, intent and realization are all specified in terms of system structures
- Note also that the pictures of devices in a network structure example diagram are essentially Occurrences.




Note that Terms are from the ONF Core Model and the above diagram is adapted from a diagram constructed by the ONF OIMT project team (https://wiki.opennetworking.org/display/OIMT/ )

# Partial Compatibility



# Target and next steps

- There does not seem to be readily available terminology to label/define the concepts (Occurrence etc.) in the problem space
  - Hence it has been difficult to discuss the properties of the language
  - **Action:** Improve terminology and definitions
- It appears that there is not a good language suited to solve this problem fully.
  - This may only appear to be the case, i.e., there may be a language out there (as it has proved very difficult to describe the problem)
  - **Action:** Continue to explore and refine
- It is possible that YANG could evolve to be more suitable
  - Current YANG does not have the necessary structures or recursion
  - A language sketch is being worked using a JSONized form of YANG to unify the class and instance statement representation
  - **Action:** Seek assistance from netmod to develop the approach and language

- For expression of:
  - Intent
  - Capability
  - Partial Visibility
  - Planning
  - Negotiation
  - Policy
  - Profile/Template
  - Occurrence
  - Etc.

# Thank you!

Questions?

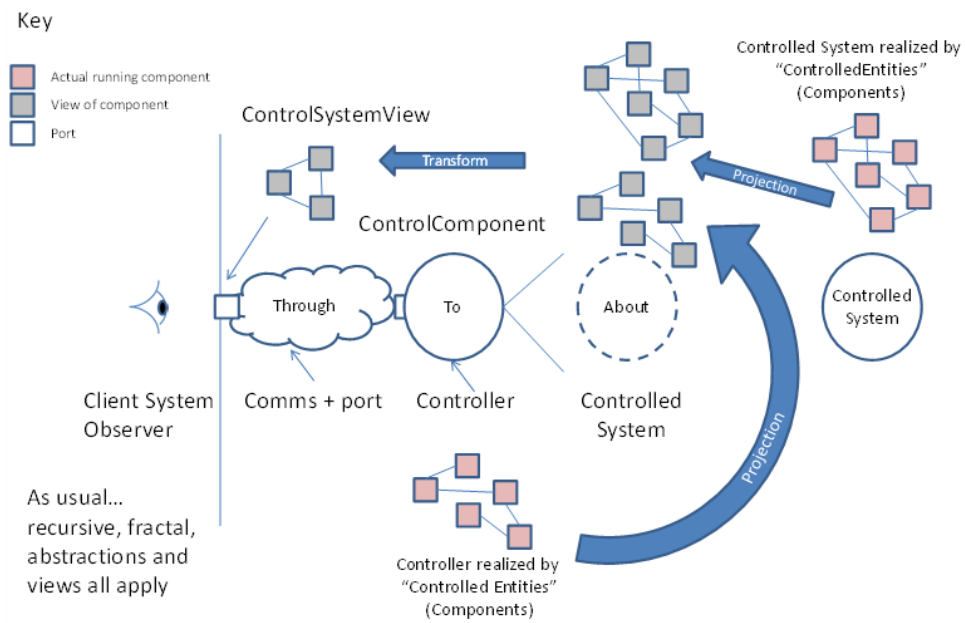
# Backup

# I-D coverage

- Progressive narrowing as a methodology (inc. narrowing/splitting/merging)
- Capability expression
- Compatibility
- Boundary (complexity and fuzziness)
- Focusses
- Partial description
- Partially satisfied request
- AI and uncertainty
- Property (discussing extension of definition)
- Two distinct perspectives and viewpoints
- Occurrence (including an expansion of the concept in an appendix)
- Intention-Expectation
- Outcome and experience
- No longer “instances”
- One model and uniformity of expression
- Foldaway complexity
- Metamodel v model
- A few thoughts on Tooling
- Problem examples (just a list at this point)
- Enhanced Yang sketch (in a JSONized form)

# Observation: Two distinct viewpoints

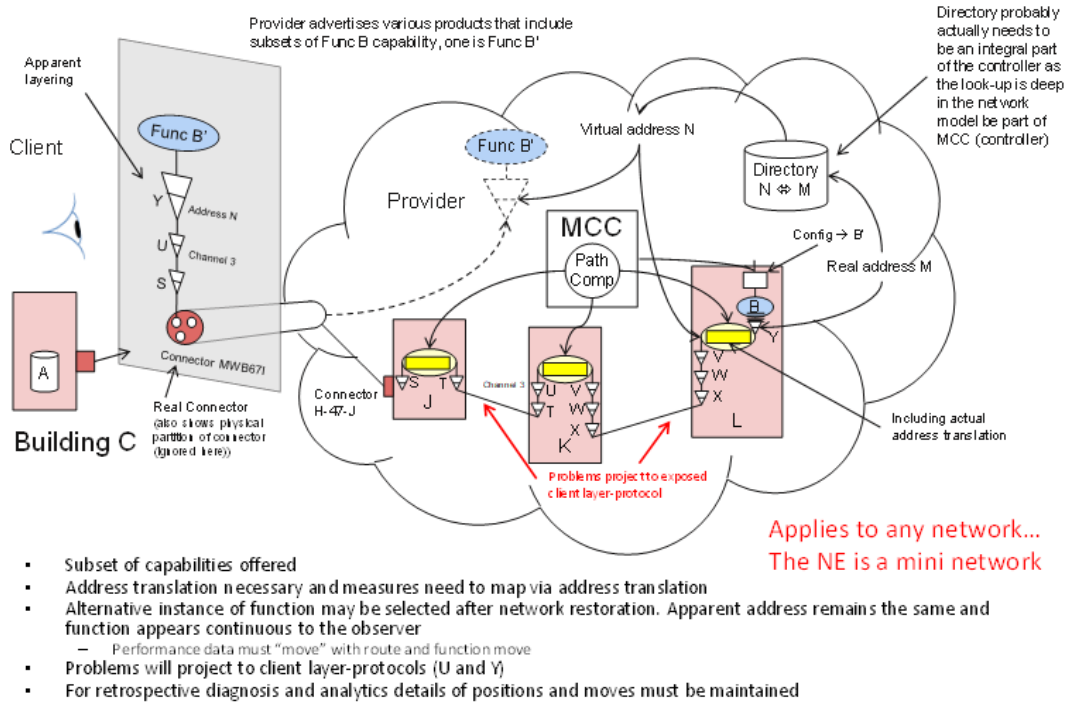
- The external perspective (the effect) – “exposed”
  - Capability (advertised to enable negotiation and selection)
  - Intent (the agreement resulting from the selection at the end of negotiation)
  - Achievement of intent
- The internal perspective (the realization) – “private”
  - Realizations (alternative system design approaches to achieve exposed capabilities)
  - Specific chosen realization (the system to be deployed)
  - Actual realization achievement
- Both viewpoints are expressed using the same metamodel
  - A Component described in terms of a System of Components
- Note that the external perspective relates to “CFS” (Customer Facing Service (TM Forum)) and the internal perspective to “RFS” (Resource Facing Service (TM Forum)), BUT the approach is used recursively throughout the entire solution
  - At any arbitrary demarcation, the same approach may be applied
  - The actual chosen demarcation may shift through evolution




Note that the above diagram is publicly available via the ONF publication at TR-512_v1.5_OnfCoreIm-info.zip

# Observations: Capability

- Capability is the expression of effect and is not the specific realization
  - It is NOT exposing intellectual property related to how the capability is achieved
  - It will include performance and cost (environmental footprint etc.) parameters etc.




Note that the above diagram is publicly available via the ONF publication at TR-512_v1.5_OnfCoreIm-info.zip

# Observation: Outcome and experience

- An outcome may:
  - Be a fixed state (first order)
  - Be a fixed change of state (second order)
  - Be a… (nth order)
  - Abide by some defined algorithm
  - Etc.
- Experience is the recipient's "perception" of the outcome
- Both outcome and experience can be expressed using the same approach discussed.
- A connectivity example outcome is an E-Line (a resource!) and the experience is apparent adjacency (the true "service"…. proximity)

# Observation: Other solution elements that benefit

Terms used here as per general industry definition (and ambiguity)

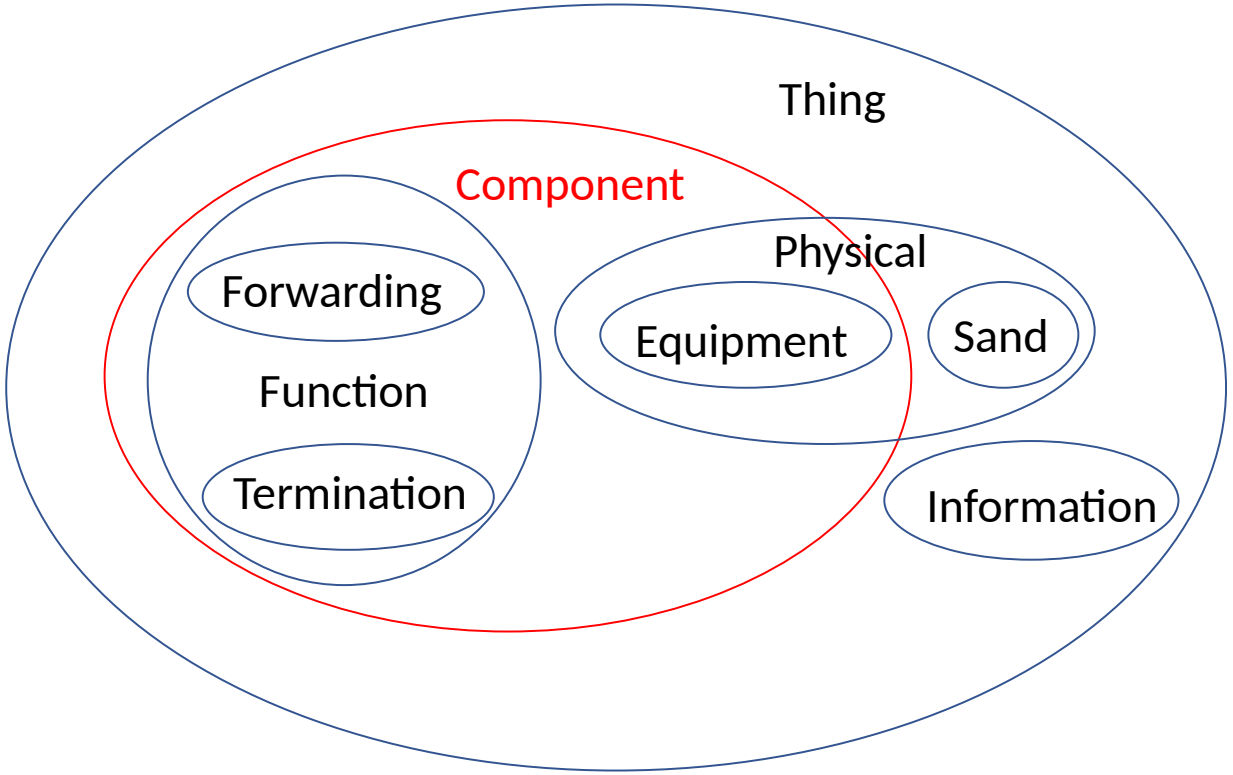
- **Policy:** The condition statement could benefit from a generalized metamodel approach to range etc.
- **Profile/Template/Redefine:** Various methods that allow application of constraints on multiple instances from a single statement. The constraint statement would benefit
- **Constraints:** In UML… An add-on that tends to be “beyond” the normal model. The essential metamodel would inherently include interaction constraints.
- Etc.

# Exploration: Focusses, boundaries…

Thing has all possible characteristics.

Specific semantics relate to the specific modelled thing and are a narrowing of thing.

The definitions do NOT need to be orthogonal/disjoint.




This presentation primarily considers Components

Consider the Termination

- Covers all aspects of “carrier” signal processing
- Coverage includes recursive definition of encapsulated forwarding
- All possible properties of termination including adaptation are within the allowed set
- Specific properties are defined in specific specifications.
- Property values are expressed in “instances”

Note that this diagram is adapted from a diagram constructed by the ONF OIMT project https://wiki.opennetworking.org/display/OIMT/

# Solution: Metamodel considerations

- Each property is specified in terms of constraints which may be narrowing of prior definitions
  - A standard may narrow an integer range
  - A usage may narrow the standard integer range
  - Etc.
- Any property, e.g., temperature, may have:
  - A detector
    - Allowing opportunity for approximate, unknown, range etc.
    - Allowing notification of change with definable approach to hysteresis etc.
  - An associated control
    - Which has intent, achievement etc.
    - Especially where it takes time to take the control action may have some progress on the action etc.
  - Have Thresholds etc.
    - Which has intent (as above)
    - Which has an associated state (allowing opportunity for approximate etc.), notification etc.
  - Have Property interrelationships for any of the above
  - Have Units for any of the above
- Where any property and its range of opportunities is stated in a specification
  - Where any invariant values in the specification are not be reported in the state of the “instance” (unless the instance is no longer behaving as defined in its specification)
- Ideally the metamodel should be such that, when a model designer chooses to define a property, they pick which of the above features are relevant and need not specify each separately.
  - Automatic name generation etc. where the name structure can be predefined.

# Various constraints

| color example illustration | Type | Constraint | Value range | Comment |
|---|---|---|---|---|
| Baseline definition | Enumeration | No constraints | red, green, mauve,… | Horrifyingly long list of arbitrary labels  |
| Intent definition | | Preferred | red | ideal |
| | | Acceptable | green | Will put up with |
| | | Mandatory when… | 6:00 – 18:00 | Outside these times color is not relevant |
| Spec definition | | Opportunities | green, blue, yellow | Specific values |
| | | Default | green | |
| R,G,B form | Numeric 3-tuple | 0-255 | 0-255, 0-255, 0-255 | |
| Translation | | | green = 0-3, 255-220, 0-3<br>etc. | Not a simple translation |

# Observation: No longer instance config…

- An “instance specification” is actually a tight statement of Intent and hence not something distinct
  - The lowest available visible view of a realization may not be precise
  - Intent has a mix of degrees of tightness of statement from vague to single value
    - The intent expression should be suitable to use for all cases including “instance specification”
  - A single expression method should enable expression of a mix of ranges and of single values
- An “instance state” is an abstraction of real state viewed through a detector
  - Detectors
    - Are imprecise
    - May fail to operate
  - The information from a detector may be
    - Temporarily unavailable
    - Delayed
    - Etc.

# The Component

Note that the diagram above was republished in ONF TR-512 with agreement from TM Forum and is publicly available via the ONF publication at TR-512_v1.5_OnfCoreIm-info.zip