

IEEE
Future
NETWORKS

Enabling 5G and Beyond



1

Modelling Boundaries

Expressing Intent, Capability and
Partial Visibility

Nigel Davis (Ciena)
20220905

The discussion in this document is an exploration of the problem.
None of the material has been agreed to any degree in IEEE.

Overall Exploration

- Abstract problem summary
- Examples of modern solution concepts
- Complexity similarities
- The nature of the solution
- Solution Metamodel considerations
- Target and next steps

There are Observations throughout this presentation
There is not time in this presentation to dig into each area

Observation: Terminology

- We all run into the challenge of terminology
- Terms are for communication convenience and not fundamental... BUT
 - Each term comes with baggage
 - Each of us has a (subtly) different understanding of each term
 - Some terms spread across a very wide space
- Each key term used in this document has specific local meaning
 - It is probable that the definitions here are too vague to ensure full shared understanding

Abstract problem and solution summary

- Current modelling techniques appear to have boundaries that make representation of some concepts in modern problems challenging
 - The concepts all have in common the need to represent uncertainty and vagueness
 - The challenge results from the rigidity of boundary representation, including the absoluteness of value and the process of classification itself, delivered by current techniques
- When describing solutions, a softer approach seems necessary
 - Intelligent control could take advantage of partial compatibilities etc. if the representation was achieved
- The solution appears to require
 - Expression of range and focus as a fundamental part of the metamodel
 - Recursive tightening of constraints as a native part of the technique

| OtsiConfigPac | |
|-----------------------|------------------|
| nonAdjacentSpectrum | : Integer [0..1] |
| spectrumBandwidth | : Integer [0..1] |
| centerFrequency | : Integer [1] |
| centerFrequencyOffset | : Integer [0..1] |
| powerSpectralDensity | : Integer [0..1] |

Examples of modern solution concepts

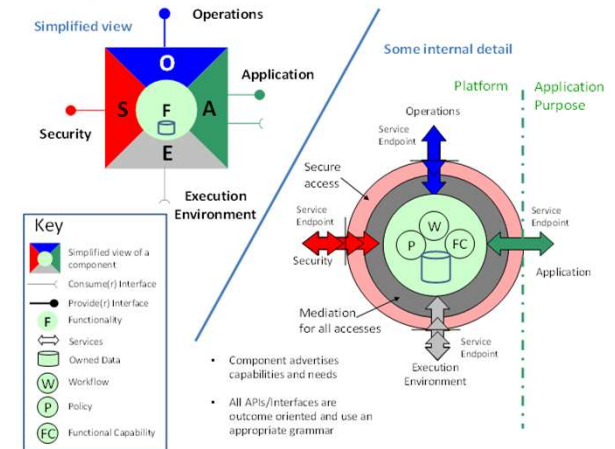
- Specification of Intent
 - Statement of desired outcome in terms of constraints
 - Includes statement of preference and acceptable value ranges etc.
- Specification of Capability
 - Statement of opportunity for behaviour to be exhibited
 - Includes statement of possible ranges and interdependencies
- 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

Observation: Progressive narrowing of Capability

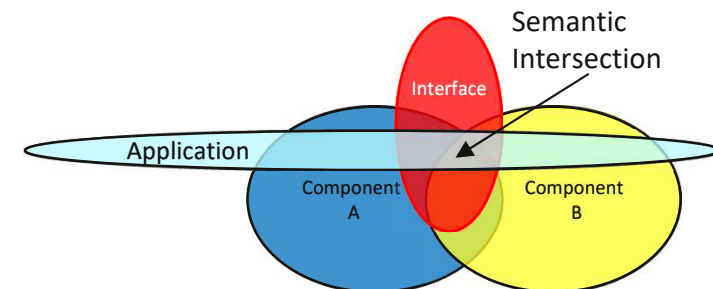
- A version of a standard may provide a definition of technology capability
 - The next version may extend/adjust that capability statement
- A vendor solution may have a narrower capability than the standard
 - The vendor solution may also extend with proprietary features
- An application of the vendor solution may have a further narrowing of capability
- A use of a vendor capability at a particular point in a structure of a solution may have a further narrowing of capability
- At a particular point in a structure of a solution under particular circumstances there may be an even narrower allowed capability
- An example or PoC of the solution may have an even narrower allowed capability
- Etc.
- Classification and statement of instance specification do not deal with the above

Observation: Compatibility

- Two components are compatible with respect to a particular application so long as their exposed capabilities have an appropriate/sufficient intersection
- Interaction of Semi-Compatible Entities using partially mappable expression
 - Semantic intersection enables a subset of capabilities
 - Some mappings are approximate and ambiguous to a degree
- The result of the intersection is usually a narrower statement of capability than the statement for the two
 - In some cases, the intersection may be the empty set
- Where a feature is preferred but not mandatory, the empty set intersection is acceptable
- Very few properties are fundamentally mandatory, importance is dependent upon specific application and operation interaction



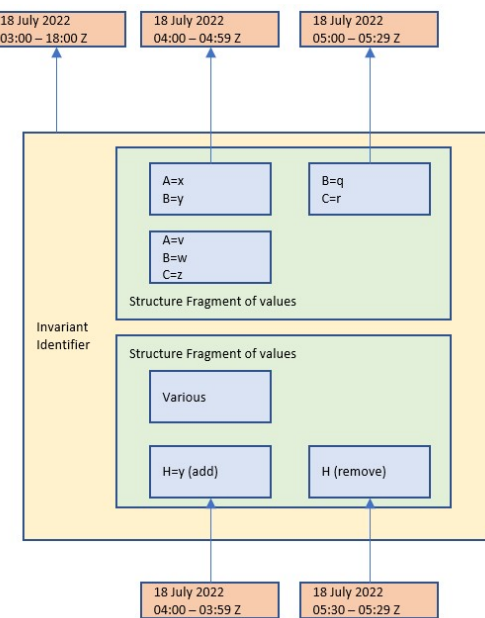
[TMFIG11118] Figure 1 The FMO component interface and structural overview



Observation: Complexity similarities

Intent, capability and partial visibility

- **Boundary**
 - Lifecycle changes: Boundary positioning/precision with maturity
 - Interdependencies between positioning of different boundaries
 - Applies to both Capability statements and Intent statements
 - Uncertainty of position of boundary and interest in positioning of boundary
 - Don't know
 - Don't care
 - Specification and measurements of acceptable, degraded and unacceptable positioning
- **Temporality**
 - Changes of positioning and precision over time
- **Probability and preference**
 - Likelihood of a particular positioning
 - Preferences for a particular positioning
- **The same challenges appear in planning and in negotiation**
 - In both there is a need to state vaguely understood and interdependent properties



The nature of the solution

- A property is stated in terms of ranges with focusses and fuzzy boundaries
 - A property statement may interrelate to statements for another property (or for other properties)
 - The statement may have multiple boundary preference level and/or probability levels
 - Note that preference and importance is per interaction and not an aspect of fundamental definition
 - The statement describes a semantic volume
 - The statement may be expressed as a narrowing of a previously expressed volume (i.e., a further narrowing)
 - A single point value is a very narrow range (many single values are actually abstractions of complex ranges, e.g. 2Mbit/s is +/-15ppm)
 - The expression should be such that the complexity of expression “folds away” for simple statements
- There is no distinction in expression opportunity between a statement of
 - Capability definition
 - Intent definition
 - Actual value
- All expressions are of the same essential form
- This is a fundamental change in the nature of the solution... a change in paradigm and metamodel
 - Fuzzy bounded/focused spaces related to other fuzzy bounded/focused spaces with preferred/probable positions etc.

Colour Intent

- preferred = red
- acceptable = green
- mandatory if for use in daytime

Colour Spec

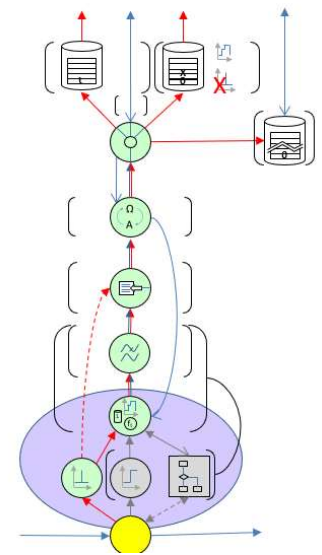
- green, blue, yellow

Colour Actual

- green (254 > value > 220)

Observation: No longer instances

- An “instance specification” is actually a tight “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.



Exploration: Rough focusses

“everything is a focused thing”
specialization → narrowing of focus

Key narrowing

- Component: ...a thing with exposed ports...
- Physical thing: ...can be measured with a ruler...

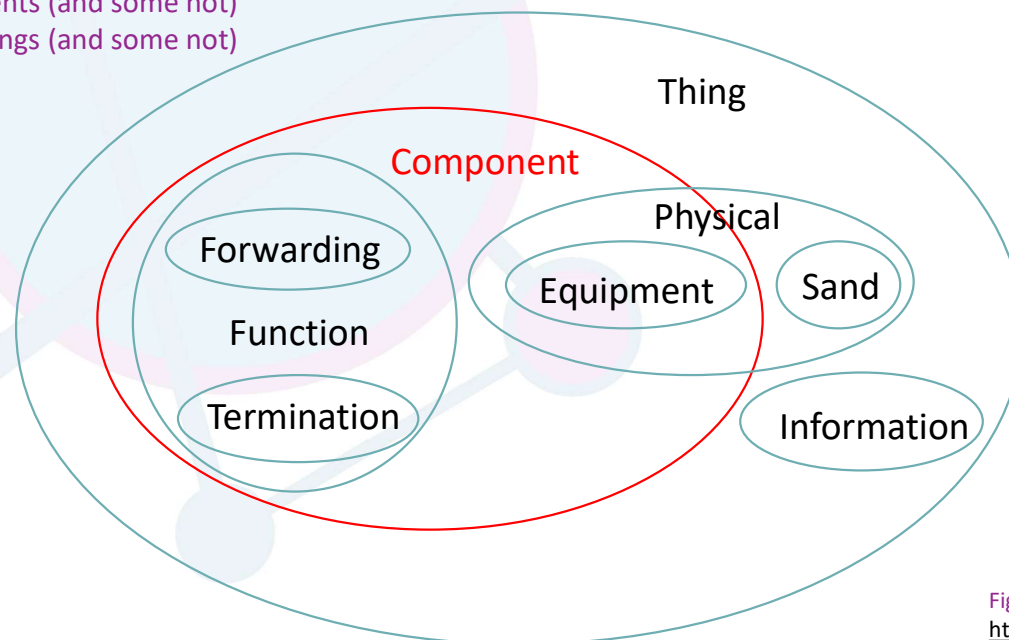
Some physical things are components (and some not)

Some components are physical things (and some not)

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.



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”

Figure from ONF work on Core Model

<https://wiki.opennetworking.org/display/OIMT/Contributions>

This presentation primarily considers Components

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” and the internal perspective to “RFS”, 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

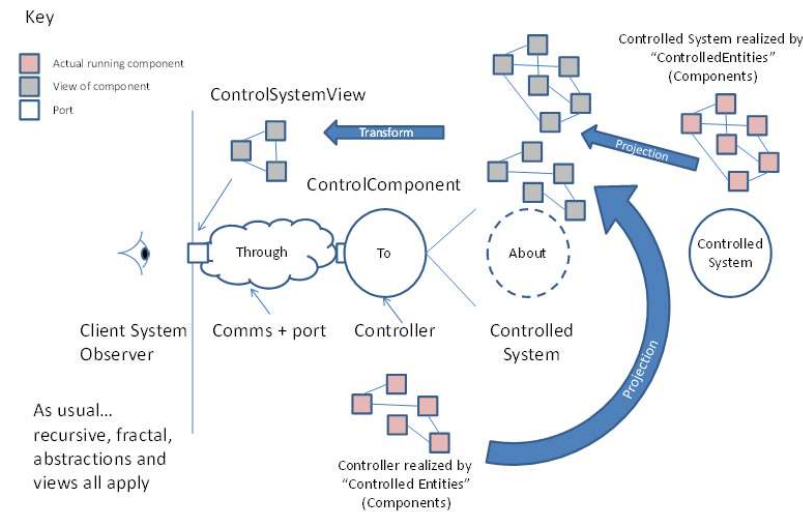


Figure from ONF TR-512.8_OnfCoreIm-Control.pdf

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.

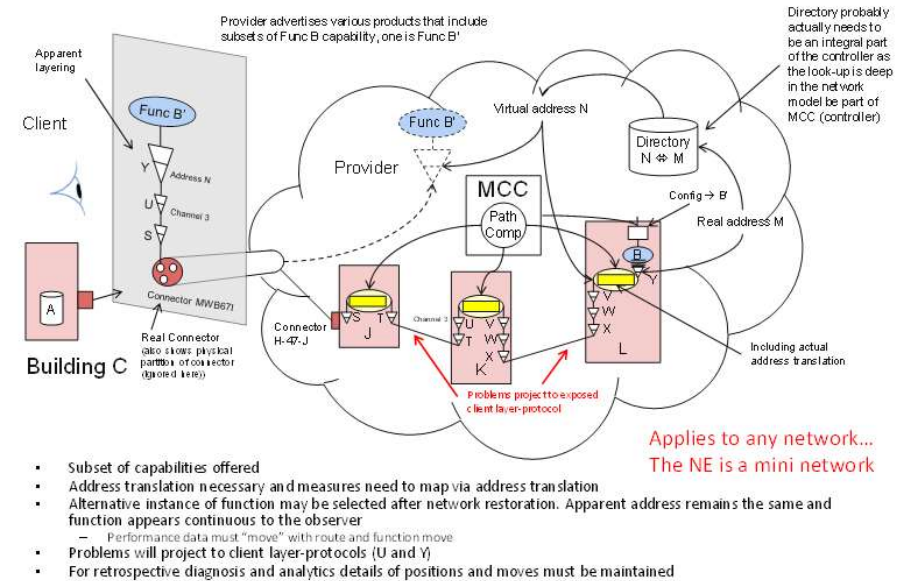


Figure from ONF TR-512.8_OnfCoreIm-Control.pdf

Observations: Occurrence

- A System structure may make repeated use of the same type of component
 - This use has been called an “Occurrence” in ONF work (see [TR-512 v1.5 OnfCoreIm-info.zip](#))
 - 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
 - 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.

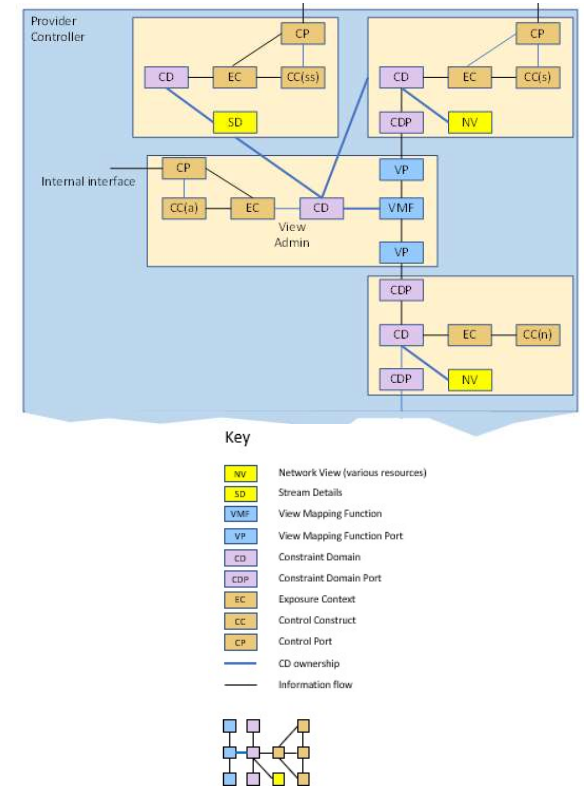


Figure from ONF work on Core Model
<https://wiki.opennetworking.org/display/OIMT/Contributions>

Observation: Other solution elements that benefit

- 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.

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 in using the same approach discussed.
- A connectivity example outcome is an E-Line (a resource!) and the experience is apparent adjacency (the true "service")

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.

Target and next steps

- There does not seem to be readily available terminology to label/define the concepts in the problem space
 - Hence it has been difficult to discuss what properties the language needs to possess.
 - **Action:** Improve terminology 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
 - Yang does not have the necessary structures or recursion
 - A proposal is being worked slowly using a JSON form of Yang to unify the class and instance statement representation
 - **Action:** Work the proposal to suitable maturity (requirements first) and take to IETF
- Note that the problem appears in expression:
 - Intent
 - Capability
 - Partial Visibility
 - Planning
 - Negotiation
 - Policy
 - Profile/Template
 - Occurrence
 - Etc.



Thank you!

Questions?