Framework for Expressing Capability

20251022

Nigel Davis

Camilo Cardona

Diego Lopez

Marisol Palmero

draft-davis-nmop-generalized-capability-principles-00 - Generalized Capability Principles

draft-davis-ivy-equipment-capability-application-00 - Equipment Capability Application

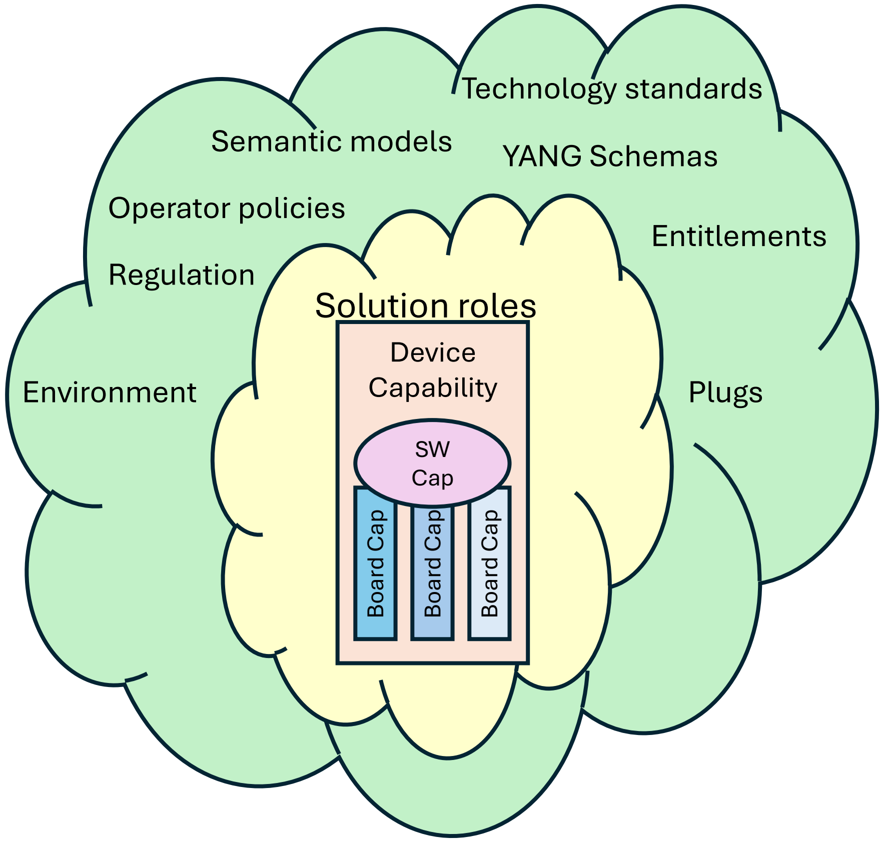
References:

ITU-T G.7711 also available at at https://opennetworking.org/wp-content/uploads/2024/02/TR-512_v1.6_OnfCoreIm-info.zip)

draft-davis-netmod-modelling-boundaries-03 - Modelling Boundaries

What is this?

- Understanding what a component can do is necessary when building any system
- How detailed? “It’s a router” is no where near enough
- Goal: A cohesive framework for expressing capability
- Approach: Develop framework in parallel with focus on specific areas delivering incremental value
- Initial specific areas: Device capability, Entitlements



A cohesive framework for expressing Capability

Thank You! ☺

Questions?

Join us for ongoing work☺

- Framework
- Equipment
- Entitlement
- Pluggable
- Green
- Etc.

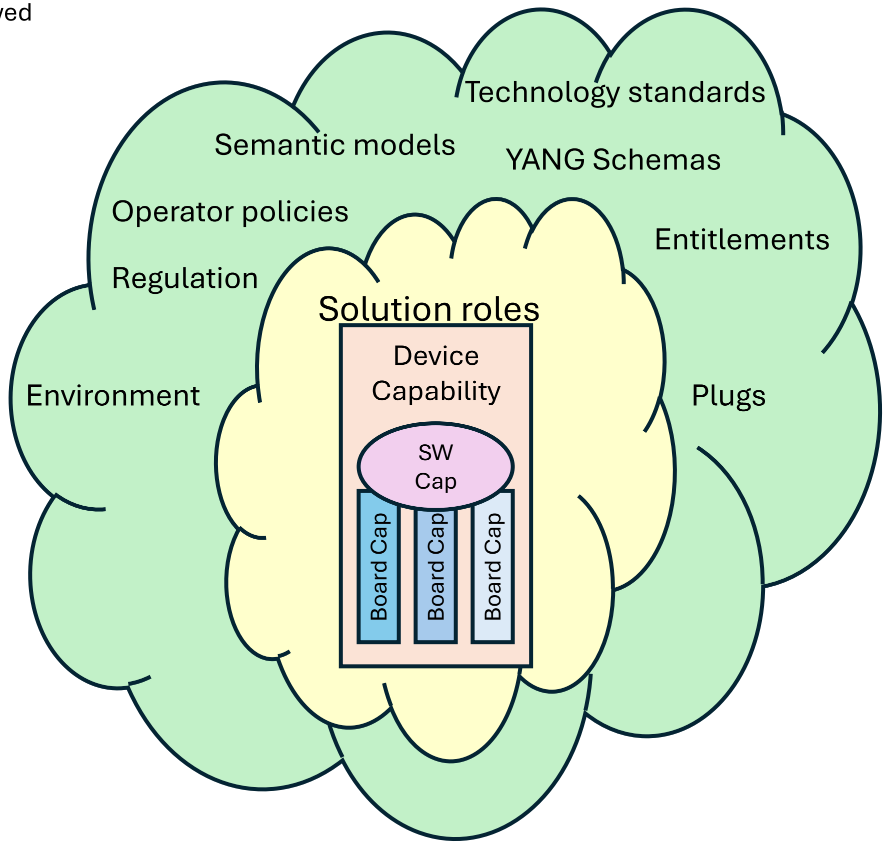
See me after if you are interested or email us:

- ndavis@ciena.com
- marisolpalmero7@gmail.com
- diego.r.lopez@telefonica.com
- camilo@gin.ntt.net

Background
and
Further Detail

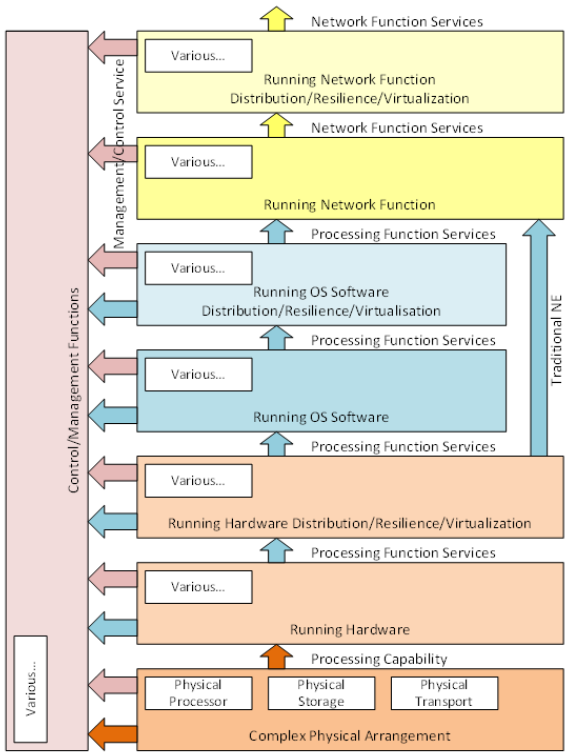
Capabilities: A use case sketch

- **Use case sketch**
  - **Purpose:** To provide a specific service
  - **Requirement:** Some combination of behaviour realizes the service
  - **Implementation:** Components (physical/functional) form system to create behaviour
    - Component-system pattern: Components can be interconnected to form a system, as system can be viewed as a component
  - **Design:** It is necessary to understand the capabilities of devices
    - To select the components
    - To analyse and adjust the design
    - RFC3535: “5. Consolidated Observations” “27. It is necessary to discover the capabilities of devices.”
- **Capabilities of a device**
  - **Today:** Capabilities described loosely in human readable text
    - Incomplete, Ambiguous, Inconsistent
  - **Challenge:** People make systems work in practice, but looseness leads to
    - Errors, Inefficiencies, Limited reuse.
  - **Next Generation:** LLMs can potentially improve interpretation
    - But when asked LLMs indicated that a rigorous model is preferrable to loose and ambiguous text
- **Goal:** A cohesive framework for expressing capability and needs
  - Expressed in a declarative, structured, and reusable form that supports recursive emergence
  - A standardized and extensible approach
    - Grounded in ITU-T G.7711 (ONF TR-512) principles
    - Based on component–system pattern (ONF_TR-512.A.2)
    - Address issues in draft-davis-netmod-modelling-boundaries-03 - Modelling Boundaries
- **Initial Focus:** Framework and Example uses (starting with ivy)
- **Note:**
  - LLMs can assist in language and framework construction and in population of capability information by mining documentation



Emergence

TR-512.6 Core Information Model – Physical Version 1.6



- Each block (Function, Software, Hardware, Physical and Control/Management) may be a single component or may be many components interacting to form one or more systems
- Each component may expose detail as if many components where the exposed components are interacting to form one or more systems
- Each set of services may be subdivided and may be complex reflecting the exposed components and their encapsulated functions
- Running OS Software is generalized to any OS including the negligible OS of a FPGA
- Distribution and resilience may not be present or may be present for only some of the services (management/control may also have distribution/resilience (not shown)
- A network function may be built directly on the Processing Function Services of the Running Hardware (i.e. is essentially a hardware realization) or some combination of running hardware and OS software
- Processing capability is a generalized term that could also apply to a parallel compute engine and a scattered set of dedicated engines such as any assembly of logic gates and buffers
- Each block is controlled/managed by a function that itself is built on Processing Capability and Processing Function Services
- The following figures use this colour scheme and often omit Management/Control

Figure 4-2 Simplified sketch of physical to functional

TR-512.6 Core Information Model – Physical Version 1.6

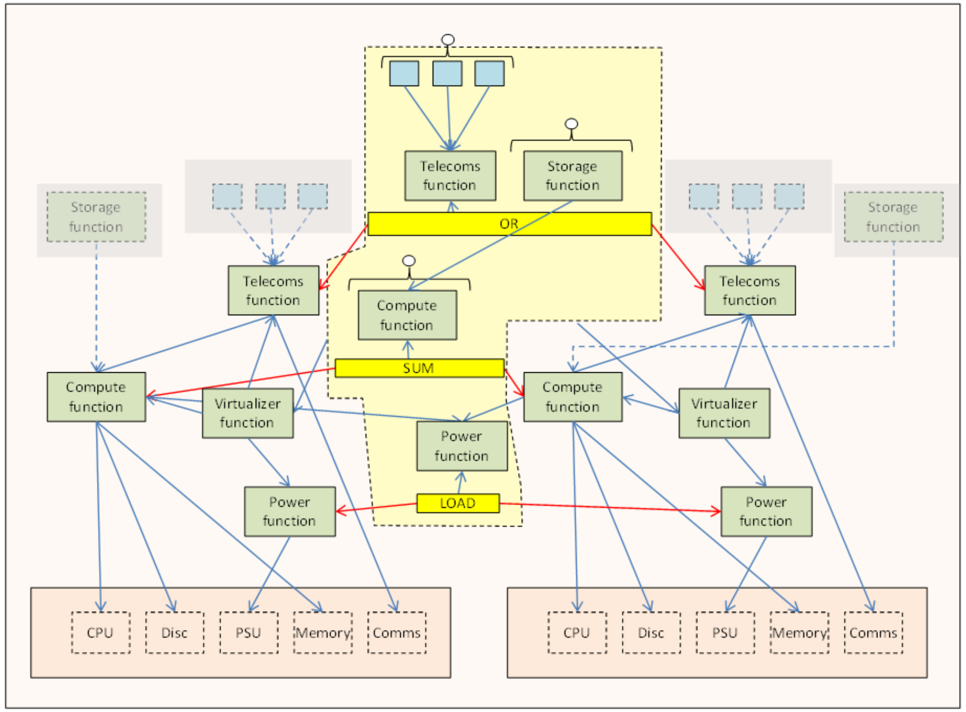


Figure 4-3 Simplified sketch of forms of resilience in an NE

Capability in an overall structure

- Capability of a component results from an intertwining of algorithmic behaviour and operating rules along with capability provided by subcontractor providers and peers

From ONF TR-512.7 (tweaked)

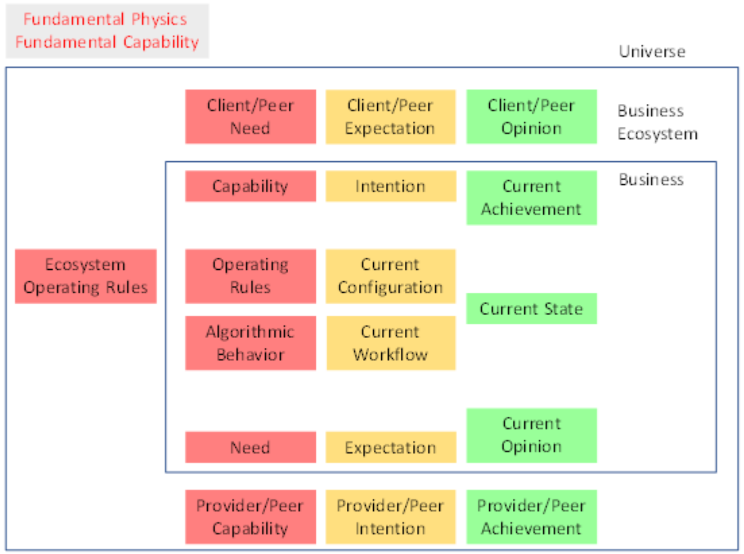


Figure 2-1 Stylized model of capabilities, intention and achievement

A framework for expression of capability

- Will support capability modelling for "any aspect" of the controlled networking solution enabling
  - Licensing and regulatory constraints
  - IVY equipment and software capability modelling
  - Photonic plug manifests development
  - Entitlement representation work
- It also supports initiatives focussing on energy/thermal
  - Power/thermal implications of detailed capabilities
- Aiming for a standardized and extensible approach for expressing features
- Grounded in ITU-T G.7711 principles
- Based on component–system pattern (see ONF_TR-512.A.2)
- Will address many issues highlighted in [draft-davis-netmod-modelling-boundaries-03 - Modelling Boundaries]
- Uses recursive functional emergence pattern
  - Repeated use of pruning & refactoring of components to form specialized narrow occurrences
  - Assembly/intertwining/interconnecting components to form systems
  - Each system is viewed as a component

Progressing the work

Challenges

- Complexity of problem
- Necessary sophistication (complexity) of solution
- Absence of a complete language for description

Mitigation

- Sketch target patterns as a vision to help direct early work
    - Component system
- Focus on applications and develop high value first
    - Device capability
- Propose Gradual progression with value justified steps
    - Soft references first
- Expect migration of languages
    - Take advantage of semantic consistency through migration and provide syntax translation description
- Ensure that any degree of capability description can be provided
    - Provide soft reference options at all staged (text as opposed to navigable link)
- Focus on examples
    - Initial application is to device capability
    - This will lead naturally to (and include) capabilities of functional components

Example from TR-512

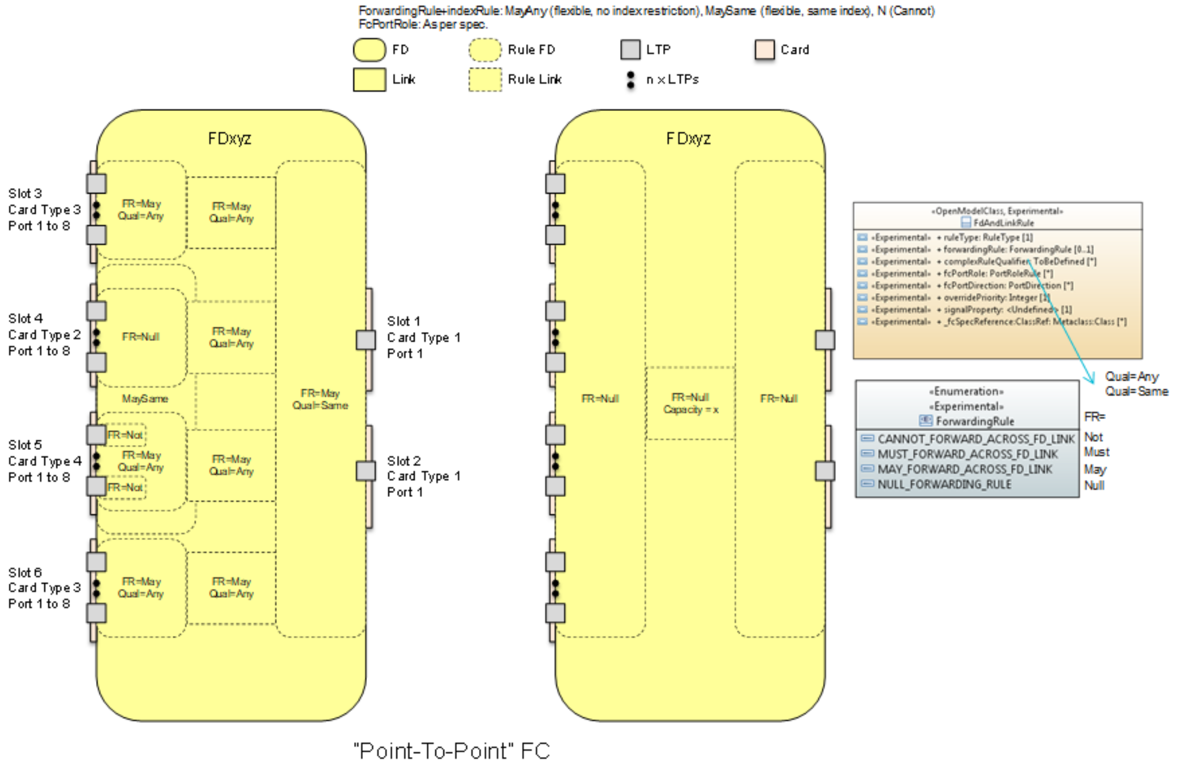


Figure 3-3 Simple FD Spec showing further overlay of rules