Clarifying the Concept of Intent

draft-clemm-nmrg-dist-intent-01

Alexander Clemm (Huawei, USA)
Laurent Ciavaglia (Nokia, France)
Lisandro Zambenedetti Granville (UFRGS, Brazil)

with input from many others:
Jeferson Campos Nobre, Mouli Chandramouli, Toerless Eckert,
Dimitri Papadimitriou, Padma Pillay-Esnault, Kaarthik Sivakumar
Status update

• Initial discussions on this at IETF 100/101 + NMRG interim at IFIP/IEEE NOMS 2018

• Per discussions, the first in a suite of eventually three drafts:
  (1) Terminology – Definitions and Concepts: Intent vs policy vs service models, etc
      \textit{This draft}

  (2) Intent definition – Expressing Intent (\textit{draft TBD})
      - Human – Machine interface aspects
      - Relationship to data models – can you use YANG?
      - Layer interdependencies

  (3) Basic intent architecture and framework/reference architecture
      \textit{draft-moulchan-nmrg-network-intent-concepts}
      - How to render intent
      - How to validate network behaves “as intended”

• Various updates from -00: editorial updates and tightening, added references
What is this about?

• “Intent-Defined Networking” is one of the recent industry buzzwords
  • Basic idea: Define what you want, not how to do it
  • This sounds good, but is this idea really new? (rhetorical question)
    • Policy-based management: Define high-level policies, leave it to policy renderers to do the rest
    • Service models and service provisioning:
      Define services & leave mapping of the service to low-level configurations, resource allocations, and objects to a flow-through provisioning system
    • Information hierarchies and abstractions are known concepts and common practice for service providers today (e.g. TMForum eTOM / Business Process Model, ITU-T TMN reference model (management layers + FCAPS)

• So, what is intent, really?
  • How does it differ from what came before?
  • Is Intent a reincarnation of policy? Of service models? Is intent synonymous, or different?
    Why all those terms and how do they relate?
  • If it is different: how so? What are the implications?
Existing Frameworks (that also make extensive use of management abstractions and hierarchies)

Source: ITU-T
Existing Frameworks (that also make extensive use of management abstractions and hierarchies)

- **Network Element Management**
  - Manage aspects of individual devices: monitor ports, download a patch, set a configuration parameter

- **Service Management**
  - Manage network connectivity aspects: topology, links, end-to-end paths

- **Network Management**
  - Manage services provided by the network: service level management, service order provisioning

- **Element Management**
  - Manage overall business aspects supported by the provided services: revenue forecast, supply chains, corporate compliance

- **Business Management**

Source: ITU-T
Existing Frameworks (that also make extensive use of management abstractions and hierarchies)

eTOM – enhanced Telecoms Operations Map
(TM Forum GB 921: Business Process Framework)
Differences between concepts and terms

- **Service Models:**
  - Describe instances of services that are provided to customers (see e.g. RFC 8309)
  - Service instantiation involves **orchestration** and **mapping** to underlying resources (user does not specify how to add, modify, remove a service – the system does it)
  - Machine-to-machine interactions; flow-through provisioning
  - Typically centralized

- **Policy:**
  - Set of rules (event/condition/action or variations)
  - Imperative: specify **how to act** / what to do under what given circumstances
  - (largely) machine-to-machine (but also devops-to-machine) interactions
  - Policy rendering: **abstraction** (and homogenization) of low-level knobs and data

- **Intent:**
  - Declarative: Define desired **outcomes** and high-level operational goals
  - Interactions between humans and machines
  - Network (or Intent-Based Management System) renders intent – two aspects: **information abstraction** and **determination of logic**
  - Centralized and decentralized flavors
Discussion items

• Define intent narrowly (only “new” concepts) or broadly
  • Putting things into a common context vs. guilty of “intent-washing”
  • Operational intent – service intent – flow intent
  • Intent at different hierarchy layers (at device/network/service level), distinguished by actor (NOC operator, user, administrator)

• Intent functional areas:
  e.g. intent fulfilment vs intent validation (or assurance?)
  • Intent compliance assessment and monitoring
  • Service assurance and service level management (“intent washing”)

• Intent levels
  • Intent at multiple levels in a hierarchy: e.g. service vs network infra
  • Intent of multiple roles: e.g. NOC operator, admin, end user
  • Intent at multiple levels of granularity: e.g. flow intent
  • Intent reconciliation, intent conflict detection

• Possible expansion of scope to intent reference architecture?
Discussion items (contd)

• Intent articulation and human-machine aspects
  • Type of actor impacts the type of interaction and interface
    • Natural language processing – infer meaning
    • Dialogue vs command
      • Dealing with under-specification (and over-specification)
      • Conflict avoidance
      • Resolution of ambiguities
  • Technical solutions are beyond scope of this particular draft, but important for distinction of what makes intent “unique”
  • “Can intent be expressed as YANG data model?”

• Beyond scope but relevant for IRTF: possible research topics
  • Human-machine interaction
  • Intent compliance assessment
  • Intent conflict detection, reconciliation, negotiation

• This is ongoing work & the discussion is just getting started
• Next step: RG adoption?
Thank you!