A Unified Management Framework for autonomic and software-defined networks

IETF 86 – 29th NMRG meeting
14 March 2013, Orlando – FL, USA

WWW.UNIVERSELF-PROJECT.EU
OVERVIEW

MOTIVATIONS
UMF IN A NUTSHELL
UMF AND SDN
STANDARDIZATION OPPORTUNITIES
MOTIVATIONS
MOTIVATIONS
PROBLEM STATEMENT

Simple facts/observations on today’s networks:
- Increasing volume of traffic
- Increasing number of devices/interactions (e.g. Machine-to-Machine)
- Increasing number of services and related QoS constraints
- (still) technology heterogeneity and legacy
- (still) technology/administrative silos

Which generates the following problematic situation and detrimental impacts:
- Complexity of distributed systems and their control/management
- Reaching the limit of current management/operation practices
  - scalability, speed, highly human–dependent
- Network capabilities under-utilization
  - worst-case/over provisioning, unused advanced features
- New service or application deployment difficulty
  - slow time-to-deploy and tedious multi-techno/vendor mapping
MOTIVATIONS

GOAL

The ultimate goal of self-managing networks is to overcome these limits by providing intelligent, adaptive, modular, and automated carrier-grade control functions for seamless, end-to-end and cross-technology interworking.

Objectives

- Multi-facet unification
  - Federation of existing architectures and unification management principles across multiple technologies
- Network empowerment
  - Embed intelligence to achieve true self-managing networks
- Industry readiness
  - Demonstrate deployability and develop migration strategies for adoption by telcos/vendors
- Trust and confidence
  - Demonstrate the reliability of every autonomic solution and develop standard testing and certification

In this context, standardization is a must!
MOTIVATIONS

CHALLENGES

- Genuine research challenges (still) exist to design and develop algorithms and mechanisms capable of replacing human operation | expertise | reasoning.

- An important and complex research challenge arises for the coordination of interactions among autonomic entities (conflict-resolution, stability assurance, multi-objective optimization)

- New solutions have to be extensively and rigorously tested and exercised on real use cases and field trials to prove their applicability in carrier-grade environments and build trust and confidence from the operators in their performance and safe behaviors.

- A unified framework is then needed to enable seamless, plug-and-play deployment and interoperable operations of the autonomic mechanisms. Designing this unified framework is a challenge in itself besides the required efforts for (pre-)standardization.

Most importantly, these four research challenges should be addressed concurrently which increases the difficulty of the task.
UMF IN A NUTSHELL
Solid, well-recognized understanding and knowledge of a specific domain, aiming at improving reuse of design expertise and productivity, facilitating the development of systems of that domain[1]
UMF IN A NUTSHELL
NETWORK EMPOWERMENT MECHANISM

Approach: The right key to the lock
- Use the relevant method to solve a concrete operational problem in a specific networking environment
- Realize a purposeful self-management function (closed control loop)

NEM = method + objective + context
- Use of Bayesian inference for fault diagnosis in FTTH networks
- Use of Genetic algorithm for interference coordination in LTE networks
- Use of Self-organizing maps for Congestion Prediction in Core IP networks

NEM = abstraction of an autonomic function
- External interfaces (called “skin” in the UMF terminology)
- Description, properties, capabilities, behavior (called “manifest” in the UMF terminology)
- Enabling to capture also interactions and relationships with other NEMs
- Providing uniform model and control means
Ecosystem diversity

- Multiple heterogeneous NEMs
- Multiple technology domains
- Multiple roles per NEM
- NEMs interact
  - Intra-domain
    ✓ Explicitly | Implicitly
  - Inter-domain
    ✓ Explicitly | Implicitly
Commonalities

- Common borders for a domain
- Same hierarchy
  - Reliable operation
  - Trustworthy interworking
  - Seamless deployment
- Same interfaces
  - Policy
  - Group communication
  - Sensing
UMF IN A NUTSHELL
UMF CORE FUNCTIONAL BLOCKS

Seamless deployment and trustworthy interworking of NEM army require:

- Tools for the operators to deploy, pilot, control and track progress of NEMs in a unified way
  - GOVERNANCE functional block

- Tools to identify/avoid conflicts and ensure stability and performance when several NEMs are concurrently working
  - COORDINATION functional block

- Tools to make NEMs find, formulate and share relevant information to enable or improve their operation
  - KNOWLEDGE functional block

- APIs to enable NEMs “plug and play” deployment, interoperability and monitoring/configuration
  - NEM Skin
  - Specific adaptors
Objective of the UMF Core:
Seamless and trustworthy deployment of NEMs

Accomplished by specification, and then standardization, of:

- Interfaces
- Coordination schemes
- Communication patterns
- Knowledge structures
- Policy translation levels
- Ontology
- Recommendations for NEM development (lifecycle, generic structure...)
UMF IN A NUTSHELL
UMF CORE FUNCTIONAL BLOCKS

Responsible for:
- The interaction between human operator and its network→express business goals report on critical states of self-managed operations/devices
- Driving NEMs’ behavior→policy-based framework for translating business-level, service specific goals/requests into low level, policies and configuration commands

GOVERNANCE ↔ NEM:
- Commands to set NEM’s status/mode (e.g. active, idle, stopped) and configure its operational parameters.
- Report on the NEM’s operational conditions and configuration characteristics (e.g. performance indicators, capabilities/behaviour, interaction with other NEMs).

IETF 86 – 29th NMRG meeting
Responsible for:

- Ensuring the proper sequence in triggering of NEMs and the conditions under which they will be invoked taking into account:
  - Operator and service requirements,
  - Needs for Conflict avoidance, joint optimization and stability control.

COORDINATION ↔ NEM:

- Commands to drive coordination including: tokens, timing, constraints, status (active/idle), etc.
- Information on the NEMs operation including: parameters, metrics, scope, utility functions, etc.
**Responsible for:**

- Providing the suitable probabilistic models methods and mechanisms for derivation and exchange of Knowledge, based on:
  - Context and configuration information from NEMs,
  - Policies from Governance,
  - Information on NEM interactions from coordination

**KNOWLEDGE ↔ NEM:**

- Commands to retrieve, share, derive and manage knowledge including: publish, subscribe, push, pull, request, store, notify ... messages.
- Registration of NEMs.
UMF IN A NUTSHELL
NEM LIFECYCLE

CREATE NEW INST.

INSTALL

DEPLOYING

REGISTERING

INSTANTIATED

DELETE

MANDATE

NEM Class (software)
described by MANIFEST (machine readable)

NEM Instance
described by INSTANCE DESCRIPTION

Life-cycle:
Detail the states and transition of a NEM instance, from its being installed, to it running its MAPE autonomic loop.

Steps include all the management by the UMF core functional blocks.

IETF 86 – 29th NMRG meeting

UMF IN A NUTSHELL
NEM LIFECYCLE
Different time scales, different events

- **MAPE of a NEM instance**
- **Control of a NEM instance by COORD**
- **Activation of a NEM instance by GOV**
- **Deployment of a NEM instance**

### Information Exchange flows
- Information Exchange flows with COORD
- with other NEMs

### Managing the NEM: Setting Up or Down
- Setting control policies to avoid conflicts e.g. disabling actions, or giving token

### Creating an Instance, Deploying it over resources/services
- Registering this instance to UMF core blocks
UMF in a nutshell

Information model

UMF information model TMF SID-compliant

- Provide formal UMF specification based on a standardized subsets of TMF SID
- Ensure coherence between implemented classes generated from IM classes
- Used to model the exchanged data and the policy structure within the governance block
- Ease UMF integration in telcos IS environment

Design approach

- UMF concepts defined and mapped to SID
- New concepts added via SID patterns e.g. NEM information model
A unified framework to deploy and control self-managing functions

- Specifications of the UMF core functional blocks
- Specifications of the NEM
- UMF and NEM APIs (skin) and workflows/sequence charts
- Publicly available specifications, developer guidelines
- Implemented, tested, modular and re-usable components
  - NEM skin
  - RESTful APIs
UMF AND SDN
UMF and SDN

UMF defines the necessary abstractions/APIs
  - for autonomic functions (NEMs)
  - from the management point of view (UMF functional blocks)

SDN is essentially about abstractions and APIs

Complementarity where the abstractions will meet
  - Starting by identifying SDN management requirements and specificities
STANDARDIZATION OPPORTUNITIES
STANDARDIZATION OPPORTUNITIES

AFI ISG

Scenarios, Use Cases, and Requirements for Autonomic/Self-Managing Future Internet

WI#1

New Scenarios, Use cases & requirements

New elements of Generic Autonomic Network Architecture

WI#2

Architectural Reference Model for Autonomic Networking

Evolution of eTOM with autonomics

AFI ISG

Technology, Evolution, and Operations Management (TEOM)

TMForum

MTOSI/MTNM

Learning for Traffic Engineering, Routing, Fault-diagnosis

Metrics, comparative research, collection of use cases

LCCN RG

Semantics, governance & coordination mechanisms

NGMN

Architecture, features, use cases, measurements and requirements for SON mechanisms and coordination, O&M aspects and requirements, system architecture and service requirements for future mobile networks

3GPP

Protocols and algorithms for managing constrained devices, Autonomics in the Internet

IRTF

IETF 86 – 29th NMRG meeting
QUESTIONS & ANSWERS
The research leading to these results has been performed within the UniverSelf project (www.univerself-project.eu) and received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 257513
PROJECT ID

- FP7 Call 5 Integrating Project
- Total Cost: ~16M€; EC Contribution: ~10M€

- 16 Partners (3 Vendors, 4 Operators, 4 Research Institutes, 5 Universities)
- Coordinator: Alcatel-Lucent
- Duration: 36 months
- Start date: 01/09/2010
- Website: www.univerself-project.eu
BACKUP SLIDES
UMF IN A NUTSHELL
CAPABILITY LEVELS

0 – Reliable operation of a standalone NEM
  o 0.1: Reliable decision making under noise
  o 0.2: ... with context awareness
  o 0.3: ... with prediction

1 – Trustworthy interworking of NEMs in a Team
  o 1.0: Orchestrated Team work with maximal utility
  o 1.1: ...with sharing of relevant context changes
  o 1.2: ...with sharing of relevant predictions

2 – Seamless Deployment of NEMs, NEM Teams
  o 2.0: NEM/NEM Team Lifecycle Management
  o 2.1: ...with governed context sharing
  o 2.2: ... with governed knowledge building
**UMF IN A NUTSHELL**

**CAPABILITY LEVEL 0.0**

---

**GOV**
- Decides based on rules when and how to act

**Sense**
- Measures, Receives, Retrieves, Estimates, Etc.
- Current values of KPI’s

**Rules**
- Thresholds, Weights, etc.

**Act**
- Increase or decrease Parameter values

**Parameters**

---

**D.Rules**: Decision rules
- **D.Rules**: IF KPI_i<T_i THEN P:=P+δ

---

Event, Date, Location
**UMF IN A NUTSHELL**

**CAPABILITY LEVEL 0.1**

- **Sense**
  - Measures, Receives, Retrieves, Estimates, Etc.
  - Current values Of KPI’s

- **Rules**
  - Thresholds, Weights, Context handling, etc.

- **Act**
  - Increase Or decrease Parameter values

- **Decision**
  - Decides when and how to act based on rules

- **GOV**
  - Any means (e.g. SNMP) to manage the D.CONFIG

- **KPI’s**

- **Contexts**

- **Noise Filtering**

- **Time, location, Power, load, Media, trust,…**

- **Event, Date, Location**

- **Operational and context data**

**Example Rule**

\[(D+C).\text{Rules::=}+\text{Context handling rules}\]

IF KPI\(_i\)\(<T_i\) & Time= Busy Hours THEN P:=P+\(\delta\)
**UMF IN A NUTSHELI**
CAPABILITY LEVEL 0.2

---

**Sense**

- Measures, Receives, Retrieves, Estimates, Predicts, Etc.
- Current & future values of KPI & context
- Operational, context and predicted data

**Rules**

- Thresholds, Weights, Context, Knowledge handling, etc.

**Decision**

- Decides when and how to act based on rules

**Act**

- Increase or decrease Parameter values

---

**GOV**

- Any means (e.g., SNMP) to manage the D.CONFIG

---

**D.CONFIG.**

- (D+C+K).Rules::=+Knowledge_handling rules
  
  IF KPI_i<T_i & Time= Busy Hours & this was OK in the past THEN P:=P+δ

---

Event, Date, Location
**UMF IN A NUTSHELL**
CAPABILITY LEVEL 1.0

- **Listen to Team interface**
- **Team behaviour Rules**, including:
  - **T.CONFIG** = (teams, timers, msg. patterns, ...)
  - **JOIN**, **LEAVE**, **PAUSE**, **SEND**, **VOTE**, ...

Any means (e.g. SNMP) to manage the **T.CONFIG**

**GOV**

**Decision**

**Sense**

**Rules**

**Act**

**Parameters**

---

(D). **T.Rules**: Decision-in-Group rules
IF KPI_i<T_i & Promised Utility Increase is the Highest in the Team THEN P:=P+δ

**T.Rules**: Team Behaviour Rules
On BOOT send JOIN(TEAM*);
IF Time=Period & KPI_i<T_i THEN SEND(TEAM*, Utility_Promise), etc.
### UMF IN A NUTSHELL

**CAPABILITY LEVEL 1.1**

- **Contexts**
- **KPI's**
- **Teams**

**Teams**
- **Sense**
- **Rules**
- **Act**

**GOV**
- Any means (e.g. SNMP) to manage the T.CONFIG

**Parameters**

**Teams**
- **JOIN, LEAVE, PAUSE, SEND, VOTE, ...**

**NEM**

**NEM**

**NEM**

---

**(D) (T+C).Rules:=Decision-in-Team rules**

IF KPI_i<T_i & Promised Utility Increase is the Highest in the Team & Ctxt=Allowed THEN P:=P+δ

**(T+C).Rules:=Team Behaviour Rules**

On BOOT send JOIN(TEAM*);
On Ctxt_Change SEND(TEAM*, Ctxt_Change);

IF Time=Period & KPI_i<T_i THEN SEND(TEAM*, Utility_Promise), etc.
**UMF IN A NUTSHELL**

**CAPABILITY LEVEL 1.2**

+ Team Context (team size; load, trust; stability, ...)

**Contexts**

**KPI’s**

**Groups**

**GOV**

Any means (e.g. SNMP) to manage the T.CONFIG

Team

**NEM**

**NEM**

**NEM**

**Listen to Team interface**

**Sense**

**Decision**

**Rules**

**Act**

**Parameters**

(D). (T+C+K). Rules := Decision-in-Team rules

IF KPI_i < T_i & Predicted Utility Increase is the Highest in the Team & Ctxt=Allowed THEN P := P + δ

(T+C+K). Rules := Team Behaviour Rules

On BOOT send JOIN(TEAM*);

On Ctxt_Change SEND(TEAM*, Ctxt_Change);

On PredictedUtility > Threshold SEND(TEAM*, PredictedUtility);

IF Time = Period & KPI_i < T_i THEN SEND(TEAM*, Utility_Promise), etc.
(D). T.G.Rules:=Decision-in-Group under Governance rules
On ROLE_1: (=Team Leader) On JOIN: SEND(Team*, Status), ...
G.Rules:=Governance Rules
GOV-NEM: START(NEM), START(Team); STOP(NEM), STOP(Team), REGISTER, ASSIGN_ROLE(), ...
(D). (T). (G+C). Rules := Decision-in-Group under Governance rules
On ROLE_1: (=Team Leader) On JOIN: SEND(Team*, Status), ..., On WATCH(Ctxt): SEND(Team*, Ctxt:=Relevant);
(G+C). Rules := Governance Rules
GOV-NEM: START(NEM), START(Team); STOP(NEM), STOP(Team),
REGISTER, ASSIGN_ROLE(), WATCH (Context), STOPWATCH (Context), ...

Event, Date, Location
**UMF IN A NUTSHELL**
**CAPABILITY LEVEL 2.2**

---

**(D),(T),(G+C,K), Rules: Decision-in-Group under Governance rules**

- On **ROLE_1**: (=Team Leader) On JOIN: SEND(Team*, Status), ..., On WATCH(Ctxt): SEND(Team*, Ctxt:=Relevant);
- On BUILD(Know): SEND(Team*, Build(Know));

**(G+C,K), Rules: Governance Rules**

- GOV-NEM: START(NEM), START(Team); STOP(NEM), STOP(Team)
- REGISTER, ASSIGN_ROLE(), WATCH (Context), STOPWATCH (Context), BUILD(Knowledge), STOPBUILD(Knowledge), ...
Components of a Message

- **Objective:** Trust in Autonomics
- **Audience:** Operators
- **Technology:** P+SL+A
- **Impact:** Certification

### NEM Certificate

<table>
<thead>
<tr>
<th>Ecosystem complexity</th>
<th>Operator Trust (UMF)</th>
<th>Operator Trust (no UMF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>New management (UMF)</td>
<td>New market</td>
<td></td>
</tr>
</tbody>
</table>

### NEM Maturity

- **0** – Reliable operation of a standalone NEM
  - 0.1: Reliable decision making under noise
  - 0.2: ... with context awareness
  - 0.3: ... with prediction

  **Problem being solved by a NEM**

- **1** – Trustworthy interworking of NEMs in a Team
  - 1.0: Orchestrated Team work with maximal utility
  - 1.1: ... with sharing of relevant context changes
  - 1.2: ... with sharing of relevant predictions

  **Problem being solved by a NEM group (ecosystem)**

- **2** – Seamless Deployment of NEMs, NEM Teams
  - 2.0: NEM/NEM Team Lifecycle Management
  - 2.1: ... with governed context sharing
  - 2.2: ... with governed knowledge building

  **Problem being solved by a UMF+NEM group (ecosystem)**

### Ecosystem complexity

| Capability | 2:++ROLE a set of connected **behaviours**, **rights** and **obligations** as conceptualised by **actors** in a network situation |

---

**Technology:** Predicates + Subjective Logic + Assessment