Creating a viable **Evolution Path** towards **Self-managing Future Internet** via a standardisable **Reference Model for autonomic network engineering**

**EFIPSANS** -> The Self-Managing Future Internet powered by the current IPv6 and Extensions to IPv6 towards IPv6++.  

**Self-Organization** as part of the bigger picture: **Self-Management**

**Ranganai Chaparadza** (Fraunhofer FOKUS)  
EFIPSANS Technical Manager  
Ranganai.Chaparadza@fokus.fraunhofer.de  
ran4chap@yahoo.com
EFIPSANS approach to Self-Management

Self-Managing Network -> nodes/devices are designed/engineered in such a way as to effect autonomicity i.e. feedback processes among the diverse functions, thereby enabling reactions in individual diverse functions of the network and of individual nodes/devices, in order to achieve and strive to maintain some well defined goals of the network.

Even the FCAPS functions become diffused within node/device architectures, apart from being part of an overall network architecture
EFIPSANS approach to Self-Management

- Some of the Functional Planes of the network would need to be re-factored or even merged.

- New concepts, functional entities and their associated architectural design principles that facilitate Self-Management at different levels of node/device and network functionality and abstractions, are required.

- It is not only about Automation through Scripting - Therefore: Autonomicity is an enabler for self-manageability of networks.

- By the fall of 2010, we envisage the emergence of an evolvable unified holistic Architectural Reference Model for Autonomic/Self-Managing Network Engineering i.e. for Self-Management within Node/Device and Network Architectures.
Functional Characteristics of the envisaged Self-Managing Future Internet

- Provide interfaces for **Network Governance**, 

- Auto-Discovery—devices discover services, capabilities of other devices and networks and are able to use the knowledge in dynamic context-aware Auto-Configuration.

- Self-Configuration,

- Self-Provisioning and Virtualization of Resources without requiring human intervention,
Functional Characteristics of the envisaged Self-Managing Future Internet

- Self-Composing (for services, protocol-stacks and network composition),

- Application and Service-Awareness, including awareness to survivability requirements of services and applications, which enables the network to react to the needs of applications and services.
Functional Characteristics of the envisaged Self-Managing Future Internet

- Autonomic Functionalities such as Autonomic Routing, Autonomic Forwarding (traffic engineering), Autonomic Mobility Management, Autonomic QoS Management, Autonomic Fault-Management, etc.

- Self-Monitoring
Whether an **Evolutionary approach** or **Revolutionary approach** could be taken towards designing the Future Internet:

- There is a requirement for a **Generic Autonomic Network Architecture (GANA)** as a unified holistic Reference Model for Autonomic/Self-Managing Network Engineering that allows for the production of “Standardizable” Specifications of Autonomic Behaviors i.e. **Self-* functions of context-aware Decision-Making-Elements (DMEs)**—potentially with **cognitive properties**, designed for the self-management of diverse networking environments.

**GANA = Architectural Reference Model for Self-Management within Node/Device and Network Architectures.**
Why the GANA Reference Model is Required

(1) To guide both the evolutionary approaches and the revolutionary/clean-slate approaches towards further architectural refinements and implementations,

(2) To establish common understanding and allow for the production of standardizable specifications of architectural functional entities and interfaces that guarantee interoperability.

EFIPSANS position: The current IPv6 and the extensibility of the IPv6 protocol framework opens the door to engineering autonomicity (self-managing properties) in systems, services and networks, and should be seen as a Starting Point towards the long-term Evolution of Networks towards fully Self-Managing Networks.
GANA as Common Reference Model

Evolutionary (incremental) based architectural refinements and realization (implementation) of GANA

Revolutionary (clean-slate) based architectural refinements and realization (implementation) of GANA

Common Unified Holistic GANA Reference Model
The Generic Autonomic Network Architecture (GANA) and its Hierarchical Control Loops Framework (HCLs):

GANA Principles

- Restrictions of the state of the art and potential improvements to be achieved by applying self-management

  - The Management Paradigms of today are based on the Relationship: **NMS** (Network Management System) \(\leftrightarrow\) **NE** (Network Element) and do not provide for the definition and implementation of **Manager** \(\leftrightarrow\) **Managed-Entity** Concepts and Relations and issues at different microscopic levels of abstractions, including within individual node/device architectures, down to the level of individual Protocols and System Functions.

  - **GANA** defines management and manageability aspects at **FOUR LEVELs** of node/device and network functionality and introduces **Autonomic Manager Components (Elements)** that are designed following **Hierarchical**, **Peering**, and **Sibling Relations** among each other and are characterised by autonomic control of their associated **Managed-Entities**, and co-operate with each other in driving the Self-Managing features of the Network(s).
GANA Principles (cont’d)

• Difference to today’s best known approaches

- **EFIPSANS** reviewed a number of approaches including clean-slate approaches (both pure and non-pure): such as, FOCALE, 4D, ANA, CONMan, Knowledge plane for the internet, etc, and concluded that non of these approaches proposes a holistic Reference Model that defines and distinguishes between diverse **Autonomic Elements/Managers** and their associated **Managed-Entities (MEs)** for different levels of abstractions within node/device architectures and network architectures.

- **GANA** is intended to be a holistic **Generic Autonomic Network Architecture** that defines the structures (**diverse Decision-Making-Elements (DMEs)** i.e. **Autonomic Elements/Managers** and their associated **Managed-Entities (MEs)**, including Interfaces between DMEs (DEs in short) of the GANA’s Decision Plane and interfaces between DMEs and their associated MEs and **Control Loops**.

SOCTRATES, E3, EFIPSANS Workshop: Self-Organisation for Beyond 3G Wireless Networks: part of ICT Mobile Summit 2009, June 9, Santander, Spain
GANA Principles (cont’d)

• Difference to today’s best known approaches

GANA is also meant to address the problems of

- (1) Complexity—by defining the Abstractions for autonomic/self-management functionality at Four Hierarchical Levels;
- (2) implementing Conflict-free decision-making-processes that drive autonomicity/self-management behaviours within a node/device and the network;
- (3) How to incorporate design principles that enable “in-network management” or “network-intrinsic management” and define constraints and boundaries for in-network management;
- (4) the kind of Perspectives offered to Users/Operators of Autonomic Networks, such as Interfaces for defining Network-level Objectives.
The Generic Model can be applied to 4 different Levels of Abstractions

Four Hierarchical Levels of abstractions for which DEs, MEs and Control-Loops can be designed in GANA.

**Level-1:** Protocol Level *(the lowest level)* by which self-management is associated with the network protocol itself *(whether monolithic or modular).*

**Level-2:** Abstracted Network Functions *(directly above the protocol(s)-level)* that abstract some protocols and mechanisms associated with a particular network function e.g. routing, forwarding, mobility management, etc—whereby we can talk about autonomic routing, autonomic forwarding, etc.

**Level-3:** Node/Device Level— the level of the node/device’s overall functionality and behaviour i.e. a node or system as a whole is also considered as level of self-management functionality.

**Level-4:** Network Level -the level of the network’s overall functionality and behaviour *(the highest level).*
Model of a Decision Element (DE)

1. Get(CapabilityDescription)
2. Get(FiniteStateMachine)
3. Get(Variable(Variable_List))
4. Get(FaultErrorFailureAlarmCausalityModel)

General_NonSensoryInformationRetrieval_Interface

Sensory_Interface

1. Get(Variable List) -- This is called on this DE (as an ME) by the Upper DE.
2. Pull(Dataspec of Data to be pulled) -- This is called on this DE (as an ME) by the Upper DE.
3. Push(Dataspec of Data to be pushed) -- This “upward call” is called on the Upper DE by this DE.

OtherInteraction_Interface

Management Interface

Decision-Element (DME/DE)

1. Start(Time),
2. Pause(Time),
3. Resume(Time),
4. Terminate(Time),
5. Enforce_Policy(PolicySpecification),
6. DeActivate_Policy(PolicySpecification), -- This causes the ME to switch to its “default” behaviour
7. Set(Variable List and New Values),
8. Set_Identifier(Identifier) -- Sets the ID of the DE

Effector_Interface

DE-PeerDE_Interface

DE-ME(s) Interface

ME-to-DE_SensoryInformationRetrieval_Interface

Messages or Calls to the Sensory Interface of the ME, e.g. Get(Variable List), Pull(Dataspec of Data to be pulled), etc., are initiated by this DE via this “interface”. Also, Responses by the ME, to the corresponding Calls (or messages sent) initiated the DE via this “interface” are received by this DE via this “interface”.

Upward Messages or Calls initiated by the ME to this DE, e.g. Push (Dataspec of Data to be pushed) are called on this DE (or received) via this interface.

ME-to-DE_Enforcer_Interface

Messages or Calls to the Enforcer_Interface of the ME, e.g. Start(Time), Enforce_Policy(Policy_Spec), etc, are initiated by this DE via this “interface”.

Also, Responses by the ME, to the Calls (or messages sent) to the Enforcer_Interface of the ME are received by this DE via this “interface”.

ME-to-DE_General_NonSensoryInformationRetrieval_Interface

Messages or Calls to the General_NonSensoryInformationRetrieval_Interface of the ME, e.g. Get(CapabilityDescription), etc., are initiated by this DE via this “interface”.

Also, Responses by the ME, to the corresponding Calls (or messages sent) initiated the DE via this “interface” are received by this DE via this “interface”.

SOCTRATES, E3, EFIPSANS Workshop: Self-Organisation for Beyond 3G Wireless Networks: part of ICT Mobile Summit 2009, June 9, Santander, Spain
Model of an ME at GANA’s lowest layer

1. Get (Variable List) - This is called on the ME by its DE.
2. Pull (DataSpec of Data to be pulled) - This is called on the ME by its DE.
3. Push (DataSpec of Data to be pushed) - This “upward call” is called on the DE of the ME by the ME.

A lowest-level Managed-Entity (ME) at GANA’s lowest-level layer.

1. Start (Time),
2. Pause (Time),
3. Resume (Time),
4. Terminate (Time),
5. Enforce_Policy (PolicySpecification),
6. De_activate_Policy (PolicySpecification), - This causes the ME to switch to its “default” behavior.
7. Set (Variable List and New Values)
Architecture View (DEs Hierarchy in a Node/Device)

Objectives, Policies from a higher level (network-level)

Main Decision Element of the Node (Node-Main-DE)

Decision Element: Routing-Management-DE for Routing Functions

Decision Element: QoS-Management-DE for QoS Management Functions

Decision Element: Dissemination-Management-DE for Information Dissemination Functions

Decision Element: Mobility-Management-DE for Mobility Management Functions

Decision Element: AutoDiscoveryAndAutoConfiguration-DE for AutoDiscovery and AutoConfiguration Functions of the Node

This Decision Element (DE), like the Fault-Management-DE should be part of the Node_Main_DE and not a standalone DE

GANA Level-2 DEs: Abstracted Network Functions e.g. Routing, Forwarding, etc.
Architecture View (Example of a DE inside a Node/Device)

Objectives/Policies inserted/communicated by the Upper level DE

The DE may expose “Views” concerning Info known only by the DE

Decision Element (QoS-Management-DE)

The Information Set(s) and the Information Suppliers (including their interfaces and properties)

The ME can be considered as a Wrapper around all the QoS related functions (mechanisms & protocols) of the node, exposing their “Views” to the DE or otherwise we could consider the DE having direct access to the QoS related functions (mechanisms & protocols)

QoS Management Functional Block i.e. the Managed Entity(ies)-ME(s)

The QoS Management Functional Block may be the one that has direct access to the QoS Mechanisms and Protocols and does the Orchestration based on the decisions enforced by the DE

Behaviour(s) or Policy(ies) to enforce (n) on the QoS Management Functional Block

Some of the Issues calling for Specifications (as depicted on the diagram)

The DE and ME are treated separately only for the purpose of producing Specifications.

SOCTRATES, E3, EFIPSANS Workshop: Self-Organisation for Beyond 3G Wireless Networks: part of ICT Mobile Summit 2009, June 9, Santander, Spain
Hierarchy, Peering and Sibling Relations between DEs

Node X

Objectives, Policies from a higher level (network-level-DE)

Main Decision Element of the Node (Node-DE)

Decision Element of an abstracted Network Function e.g. Routing

Decision Element intrinsic to a Routing Protocol e.g. OSPF

Node Y

Objectives, Policies from a higher level (network-level-DE)

Main Decision Element of the Node (Node-DE)

Decision Element of an abstracted Network Function e.g. Routing

Decision Element intrinsic to a Routing Protocol e.g. OSPF

The Interfaces calling for Specifications

GANA’s lowest level/layer MEs: Protocols, Protocol Stacks, Services/Applications and fundamental mechanisms

Example interaction between Sibling Decision Elements
Today's protocols are complex and hard to manage because they have inbuilt autonomous decision logic which complicates manageability and causes unwanted and difficult to understand emergent behaviour among protocol interactions. Future protocols need to be simpler (with no inbuilt autonomous decision logic), and supporting the concepts of managed connectors and other concepts proposed by CONMan. Today's protocols could be evolved towards simpler protocols with no inbuilt autonomous decision logic, but should rather be basic functions implemented as manageable protocol modules with properties proposed by CONMan. Newly designed protocols should support the Model of an ME that is managed protocol proposed for future GANA protocols.
Architecture View (Example instantiation of GANA: Autonomic Routing in a fixed Net)

This cloud represents an overlay or logically centralised DE(s)

1. With wider network-wide view to perform sophisticated decisions e.g. network optimization.
2. Centralized to either avoid processing overhead in managed nodes or scalability and/or complexity problems with distributed decision logic in network elements
3. The Elements in this cloud may be the ones that provide an interface for a humans to define Goals and Objectives or Policies e.g. Business Goals

Objectives, Policies from a higher level (network-level)

Decision Element of the Node (Node_Main_DE)

Routing_Management_DE

Only the kind of Information Set(s) required for local control-loop reaction

Control Loop (Controls the "Routing"

Routing Protocol(s) and Mechanisms of the Node e.g. OSPFv3, BGP-4

Other Information Sources/Suppliers

Exposing "Views"

Other Network-Level DEs

Network-Level QoS_Management.DE

Network-Level Forwarding Management.DE

IPv6-based Network
Architecture View (Example instantiation of GANA: Forwarding and Autonomicity)

This cloud represents an overlay or logically centralised DE(s)

1. With wider network-wide view to perform sophisticated decisions e.g. network optimization.
2. Centralized to either avoid processing overhead in managed nodes or scalability and/or complexity problems with distributed decision logic in network elements
3. The Elements in this cloud may be the ones that provide an interface for a human to define Goals and Objectives or Policies e.g. Business Goals

Objectives, Policies from a higher level (network-level)

Decision Element of the Node (Node_Main_DE)

Forwarding Management_DE

Only the kind of Information Set(s) required for local control-loop reaction

Control Loop (Controls the Data Plane)

Other Information Sources/Suppliers

Exposing “Views”

GANA Data Plane Protocols and Mechanisms: IPv6 Forwarding, L2.5 Forwarding, L3 Forwarding, L3 Switching, L2 Switching, etc.
Auto-Configuration - Scenario

1. Discovers ONIX, gets authenticated, advertises the node’s point of attachment to the network, interface(s) address information and Capabilities to both ONIX and its on-link neighbours. Queries the ONIX for the address of the NET_LEVEL_RM.DE.
2. ONIX provides the address of the NET_LEVEL_RM.DE.
3. Establishes Trust with the NET_LEVEL_RM.DE. Requests for authorization to get Routing Profiles (routing configuration data).
4. Node authentication is accepted. Authorization is provided. Trust Model is established.
5. Trust Model is pushed to ONIX. Trust model contains the authorizations for the type of data that the node is allowed to get from ONIX.
6. Routing Profiles (and possibly other routing configuration data) is passed from ONIX to the FUNC_LEVEL_RM.DE through the NODE_MAIN.DE.
7. The FUNC_LEVEL_RM.DE performs the self-configuration for OSPF related parameters.
Meta-modelling & Modelling Tool-chain
Overview of Objectives

- **Establishment of GANA meta-model**
  (formalised DE structure & behaviour specification)
- **Production of GANA scenario models**
  (showcasing control loop interaction between DEs)
- **Modelling Tool-chain design**
  (integration of modelling, verification & generation tools)
- **Methodology for Modelling Tool-chain**
  (development process to operate the modelling tool-chain)
GANA meta-model

- Formalised specification of GANA architecture in line with MDA principles
- Encompasses both structural and behavioural concepts of DE design
- Enables effective validation and verification of entangled control loops
GANA meta-model (cont’d)

- Still in progress (snapshots in D1.2, D1.4)
Modelling Tool-chain

- Realises MDA principles in practice
- Glues tools of various modelling steps (meta-modelling, modelling, verification, code-generation)
- Facilitates effective collaboration among partners (expert knowledge is leveraged and integrated)
- Automates GANA architecture building
- Encourages adoption of GANA principles
Exposing the Features in IP version Six Protocols that can be exploited/extended for the purposes of designing/building Autonomic Networks and Services (EFIPSANS)
GANA as the guide/driver to the Evolution Path

**EFIPSANS Vision:** Produce standardizable, protocol-agnostic Autonomic Behaviour Specifications (ABs) for selected diverse networking environments; then use the ABs to create and drive an evolution path for today’s Networking Models, Paradigms and Protocols, in particular IPv6, towards Autonomic Networking.

**Specifications of “protocol-agnostic” Autonomic Behaviours (ABs):**

- Control Loops and their interactions within an autonomic node, as well as the interacting distributed control loops of an autonomic network.
- Control Loops at 4 levels: Network-level distributed Control Loop(s) governing the behaviour of the network; Node-level Control Loop governing the behaviour of a node; Control Loops at specific Network-Function levels e.g. a control loop of the Routing Function OR the Fowarding Function of a node; down to -- Protocol intrinsic control loops (the lowest level).

**Extensions:**

- Horizontal Extensions: e.g. New IPv6 extension headers AND/OR additional protocol fields;
- Vertical Extensions: e.g. Enhanced inter-layer interactions among IPv6 protocols, and between IPv6 protocols and other protocols of the stack, cross-layering.
- Component-wise Extensions; and New Algorithms OR add-on mechanisms.

**Evolution of today’s Networking Models, Paradigms and Protocols towards Autonomic Networking:**

- The role of the current IPv6, EFIPSANS defined complementary extensions to IPv6 (IPv6++) and Network-Architectural Extensions in engineering autonomic networks and services.

**The ABs Specifications drive the Bottom-Up approach**
In EFIPSANS some ideas on Propositions for Extensions to IPv6 are now emerging as early drafts:

- New IPv6 Extension Headers (new IPv6 protocols that complement existing IPv6 protocols).
- New Protocol Options in the Extension Headers that support the notion of Options e.g. Hop-By-Hop Options.
Exposing Features in IPv6 protocols and Propositions for Extensions to IPv6 protocols (towards IPv6++)

- **[Under Investigation]: Extensions to the “management interfaces” of some protocols to ensure enriched autonomic control of the protocols by associated Decision-Elements (DEs),**

- **[Under Investigation]: network architectural extensions such as cross-layering, etc.**
Exposing Features in IPv6 protocols and Propositions for Extensions to IPv6 protocols (towards IPv6++)

- Examples of IPv6 protocol extensions for self-managing networks being proposed by EFIPSANS include:
  - **ICMPv6++** for advanced control information exchange between Decision Elements,
  - **ND++** for advanced Auto-Discovery,
  - **DHCPv6++** for advanced Auto-Discovery,
  - some recommendations for Extensions to protocols like OSPFv3,