ETSI GR ENI 017 V2.2.1 (2024-06)

Experiential Networked Intelligence (ENI); Overview of Prominent Control Loop Architectures

Disclaimer

The present document has been produced and approved by the Experiential Networked Intelligence (ENI) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
RGR/ENI-0017v221_Ctrllloop Arch

Keywords
cognitive, control

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from:
https://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status.
Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

If you find a security vulnerability in the present document, please report it through our
Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) Experiential Networked Intelligence (ENI).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

The present document specifies a high-level functional abstraction of the ENI System Architecture in terms of Functional Blocks and External Reference Points. This includes describing how different classes of systems interact with ENI. Processes, models, and detailed information are beyond the scope of the present document.

1 Scope

The purpose of the present document is to provide further information on prominent control loop architectures that can be used in modular system design. This will be applied to the ENI reference system architecture (and any other applicable ETSI reports and standards). The present document will emphasize control loops that are adaptive and cognitive. In release 2, the present document will provide further precisions on clause 4.4.4. It will also make any other updates that are required.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GS ENI 005 (V3.1.1): "Experiential Networked Intelligence (ENI); System Architecture".

[i.2] Strassner, J., Agoulmine, N., Lehtihet, E.: "FOCALE - A Novel Autonomic Networking Architecture", ITSSA Journal 3(1), 64-79, 2007.

[i.3] Boyd, J. R.: "The Essence of Winning and Losing", June 1995.

[i.4] Strassner. J.: "Knowledge Representation, Processing, and Governance in the FOCALE Autonomic Architecture", book chapter, 2011, Elsevier.

[i.5] Strassner, J.: "Policy-Based Network Management", Morgan Kaufman, ISBN 978-1558608597, September 2003.

[i.6] MEF 78.1: "MEF Technical Specification: MEF Core Model", Strassner, J., editor, July 2020.

[i.7] IBM® Autonomic Computing White Paper: "An architectural blueprint for autonomic computing".

[i.8] Strassner, J., van der Meer, S., Won-Ki Hong, J.: "The design of an Autonomic Element for managing emerging networks and services", International Conference on Ultra Modern Telecommunications, 2009.

[i.9] Minsky, M.: "The Society of Mind", Simon and Schuster, New York, 1988.

[i.10] R. Mitchell, J. McKim: "Design by Contract, by Example", Addison-Wesley, 2001, ISBN 0201634600.

[i.11] S. van der Meer: "Architectural Artefacts for Autonomic Distributed Systems - Contract Language", in 6th IEEE Workshop on Engineering of Autonomic and Autonomous Systems (EASe), April 14-16, 2009.

[i.12] ETSI TS 103 195-2: "Autonomic network engineering for the self-managing Future Internet (AFI); Generic Autonomic Network Architecture; Part 2: An Architectural Reference Model for Autonomic Networking, Cognitive Networking and Self-Management".

[i.13] ETSI GR ENI 016: "Experiential Networked Intelligence (ENI); Functional Concepts for Modular System Operation".

[i.14] Clark, D.D., Partridge, C., Ramming, J.C., and Wroclawski, J.T.: "A Knowledge Plane for the Internet", August 2003.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GS ENI 005 [i.1] and following apply:

abstraction: hiding of unnecessary details to focus on data and information that is relevant for defining a particular concept or process

administrative domain: domain that employs a set of common administrative processes to manage the behaviour of its constituent Entities

NOTE: This is based on the definition in [i.6].

agent: computational process that implements the autonomous, communicating functionality of an application

architecture: set of rules and methods that describe the functionality, organization, and implementation of a system

cognition: process of understanding data and information and producing new data, information, and knowledge

cognition model: computer model of how cognitive processes, such as comprehension, action, and prediction, are performed and influence decisions

context: collection of measured and inferred knowledge that describe the environment in which an entity exists or has existed

data model: representation of concepts of interest to an environment in a form that is dependent on data repository, data definition language, query language, implementation language, and/or protocol

NOTE: This definition is taken from [i.6].

decision making: set of processes that result in the selection of a set of actions to take from among several alternative possible actions

domain: collection of Entities that share a common purpose

NOTE: Each constituent Entity in a Domain is both uniquely addressable and uniquely identifiable within that Domain. This is based on the definition of an MCMDomain in [i.6].

entity: object in the environment being managed that has a set of unique characteristics and behaviour

NOTE: Objects are represented by classes in an information model.

formal: study of (typically linguistic) meaning of an object by constructing formal mathematical models of that object and its attributes and relationships

information model: representation of concepts of interest to an environment in a form that is independent of data repository, data definition language, query language, implementation language, and protocol

NOTE: This definition is taken from [i.6].

inferred knowledge: knowledge that was created based on reasoning, using evidence provided

knowledge: analysis of data and information, resulting in an understanding of what the data and information mean

NOTE: Knowledge represents a set of patterns that are used to explain, as well as predict, what has happened, is happening, or is possible to happen in the future; it is based on acquisition of data, information, and skills through experience and education.

learning: process that acquires new knowledge and/or updates existing knowledge to optimize a function using sample observations

logic: formal or informal language that evaluates a conclusion based on a set of premises

management domain: domain that uses a set of common Policies to govern its constituent Entities

NOTE: A Management Domain refines the notion of a Domain by adding three important behavioural features:

1) it defines a set of administrators that govern the set of Entities that it contains;
2) it defines a set of applications that are responsible for different governance operations, such as monitoring, configuration, and so forth;
3) it defines a common set of management mechanisms, such as policy rules, that are used to govern the behaviour of MCMManagedEntities contained in the MCMManagementDomain.

This is based on the definition of an MCMDomain in [i.6].

model: representation of the entities of a system, including their relationships and dependencies, using an established set of rules and concepts

Model-Driven Engineering (MDE): approach in which models are central to all phases of the development and implementation processes

ontology (for ENI): language, consisting of a vocabulary and a set of primitives, that enable the semantic characteristics of a domain to be modelled

policy: set of rules that is used to manage and control the changing and/or maintaining of the state of one or more managed objects

semantics: study of the meaning of something (e.g. a sentence or a relationship in a model)

situation: set of circumstances and conditions at a given time that may influence decision-making

situation awareness: perception of data and behaviour that pertain to the relevant circumstances and/or conditions of a system or process, the comprehension of the meaning and significance of these data and behaviours, and how processes, actions, and new situations inferred from these data and processes are likely to evolve in the near future

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AFI	Autonomic Future Internet
AI	Artificial Intelligence
AMC	Autonomic Management and Control
API	Application Programming Interface
COM	Control, Orchestration and Management
COMPA	Control, Orchestration, Management, Policy and Analytics
DE	Decision Element
DEN	Directory Enabled Networks
DENON	Directory Enabled Networks ONtology
EMS	Element Management System

ESB	Enterprise Service Bus
FB	Functional Block
FOCALE	Foundation - Observe - Compare - Act - Learn - rEason
FSM	Finite State Machine
GANA	Generic Autonomic Networking Architecture
IBM®	International Business Machines
KP	Knowledge Plane
MAPE	Model-Analyse-Plan-Execute
MAPE-K	Model-Analyse-Plan-Execute-Knowledge
MBT	Model-Based Translation
MBTS	Model-Based Translation Services
MDE	Model-Driven Engineering
ME	Managed Element
NE	Network Element
NMS	Network Management System
ONIX	Overlay Network for Information eXchange
OODA	Observe-Orient-Decide-Act
OSS	Operational Support System
QoE	Quality of Experience
QoS	Quality of Service
XML	eXtensible Markup Language

4 Prominent Control Loop Architectures

4.1 Introduction

Most control loop architectures for adaptive and cognitive systems use both feedback (and feedforward) mechanisms. These control loop signals play a critical role in not just stabilizing the system, but more importantly, providing mechanisms for the system to learn experientially.

EXAMPLE: A simple feedback loop consists of taking past interactions with the environment and combining them with current information to guide current and future interactions.

4.2 Definition

A control loop is a mechanism that senses the performance of an object or process being controlled to achieve desired behaviour. ENI is concerned with different types of closed control loops, where the controlling action is dependent on feedback from the object or process being controlled. In other words, closed loops use feedback to monitor and adjust the behaviour of a system to achieve one or more goals.

4.3 Types of Control Loops

4.3.1 Open

An open control loop is a control loop whose controlling action is independent of the output of the object or process being controlled. This type of control loop does not link the control action to the object or process being controlled (it simply continues to apply the control action). This type of control loop will likely not be used in the ENI system.

4.3.2 Closed

A closed control loop is a control loop whose controlling action is dependent on feedback from the object or process being controlled. This type of control loop measures the difference between the actual and desired values of a set of variables to adjust a set of parameters to change the behaviour of the system to bring the actual value closer to that of the desired value.

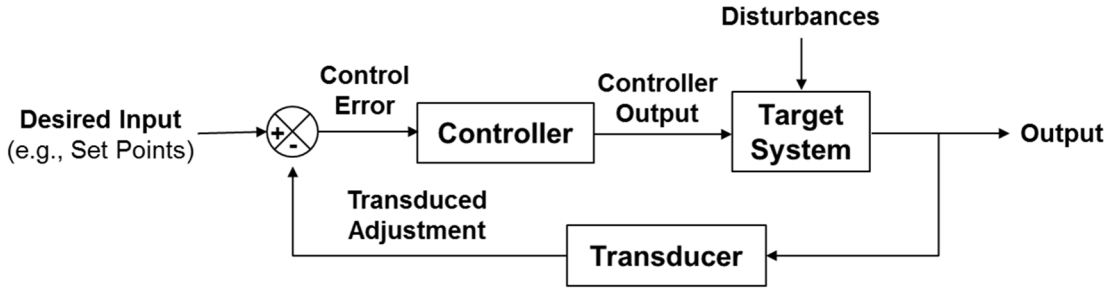

Figure 4.3.2-1: An Exemplary Closed Control Loop

4.3.3 Hierarchical Closed

A hierarchical closed control loop is a control loop that is organized in the form of a tree. This organization enables different decisions to be made by different nodes in the tree. In general, there is a set of supervisory closed control loops that allocate tasks to subordinate closed control loops. Each subordinate closed control loop performs its tasks and returns its result to its superordinate closed control loop. Advanced examples enable one of a group of designated closed control loops to take control of the hierarchy dependent on goals and the environment. This is an example of a self-organizing hierarchical closed control loop.

In general, the topmost closed control loop reasons about an abstract world model; its subordinate closed control loops reason about increasingly more specific models, or portions of models.

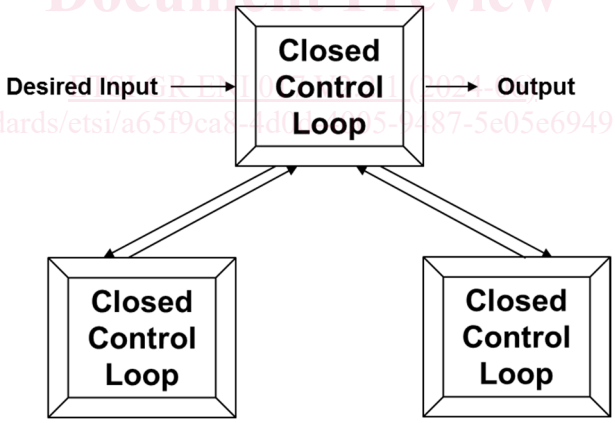

Figure 4.3.3-1: An Exemplary Hierarchical set of Closed Control Loops

4.3.4 Distributed Closed

A distributed closed control loop is a closed control loop whose components are physically distributed among different locations. Each component in a distributed closed control loop uses a message passing mechanism to communicate with one or more other components of the distributed closed control loop.

4.3.5 Adaptive Closed

An adaptive closed control loop is a control loop whose controlling function adapts to the object or process being controlled using parameter that are either unknown and/or vary over time. The parameters may be defined using a model that defines the desired closed loop performance, or statistical analysis to build a mathematical model from measured data.

4.3.6 Federated Closed

A federated closed control loop is a set of semi-autonomous closed control loops that use formal agreements to govern their interaction and behaviour. This includes rules to admit new members of the federation, as well as rules governing the visibility and types of information that can be shared with other members of the federation. Each closed control loop operates on the same goal using its own local data. Decisions from each closed control loop are then aggregated and published.

4.3.7 Cognitive Closed

Cognition is the process of understanding data and information and producing new data, information, and knowledge. A cognitive closed control loop selects data and behaviours to monitor that can help assess the status of achieving a set of goals, and produce new data, information, and knowledge to facilitate the attainment of those goals.

4.4 Prominent Control Loop Architectural Styles

4.4.1 OODA

Col. John Boyd's control loop [i.3], [i.4] and [i.5] consists of four phases:

- Observe, Orient, Decide and Act (OODA).

It is shown in Figure 4.4.1-1 which is drawn to emphasize how orientation shapes observation, decision, and action. While the loop appears to be sequential, this is merely for convenience. The orientation step is critical, as it determines how observations, decisions, and actions are performed. Hence, observation, orientation, and action occur simultaneously and continuously. As Boyd observed, people act according to how they perceive the world, as opposed to how the world really is.

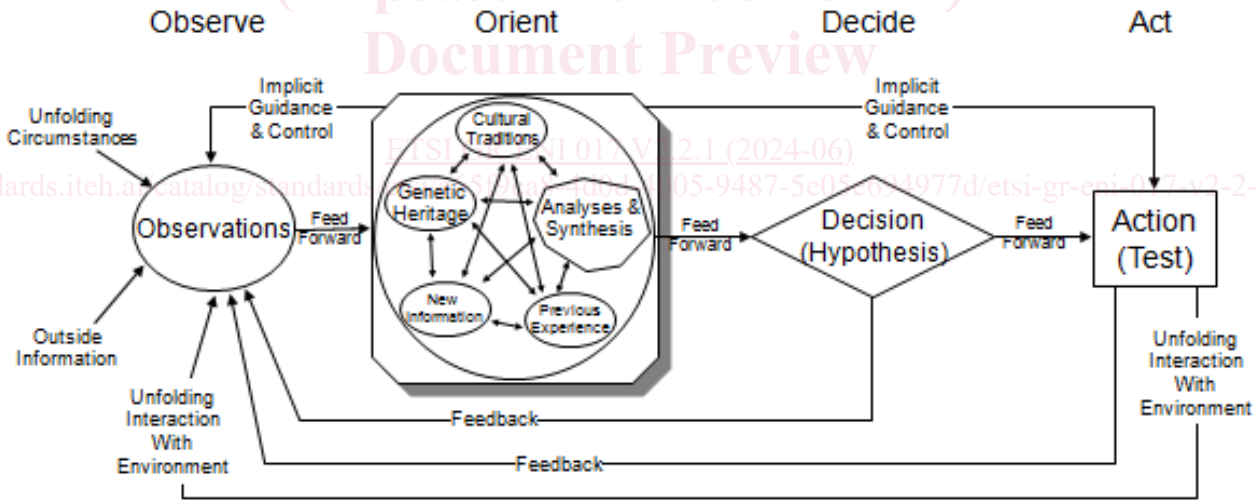

Figure 4.4.1-1: The OODA Control Loop

One of the strongest features of the OODA loop is to initiate or modify actions in response to observed events. If this can be transformed into a machine-understandable form, then formal logic can be applied to examine all different concurrent options to arrive at the best plan to achieve the goals of the mission. This is implemented in FOCALE, which stands for Foundation - Observe - Compare - Act - Learn - rEason; it is an adaptive and cognitive control loop (see [i.2] and [i.4]).

In stark contrast to other control loop architectures, OODA is a set of *interacting* loops, where observations in the current context are filtered (the orient phase) to make them relevant.

The OODA loop was the inspiration and foundation for FOCALE, which is an enhanced version of OODA that features the addition of cognition.