

ANIMA  
Internet-Draft  
Intended status: Informational  
Expires: September 22, 2016

L. Ciavaglia  
P. Peloso  
Nokia  
March 21, 2016

Knowledge Exchange in Autonomic Networks  
draft-ciavaglia-anima-knowledge-00.txt

Abstract

This document describes a solution to manage the exchange and processing of information and knowledge between autonomic functions. The objective is to provide a unified interface to enable an interoperable management of information flows among autonomic functions by insuring the use common mechanisms. The protocol negotiate and automatically adapt to the communication and information capabilities, requirements and constraints of the participating entities.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 22, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

This document may not be modified, and derivative works of it may not be created, except to format it for publication as an RFC or to translate it into languages other than English.

## Table of Contents

1. Introduction . . . . .	2
2. Knowledge Exchange Interface . . . . .	3
2.1. Information Collection and Dissemination - ICD . . . . .	4
2.2. Information Storage and Indexing - ISI . . . . .	5
2.3. Information Processing and Knowledge Production - IPKP . . . . .	5
2.4. Information Flow Establishment and Optimization - IFEO . . . . .	7
3. Acknowledgments . . . . .	12
4. IANA Considerations . . . . .	13
5. Security Considerations . . . . .	13
6. References . . . . .	13
6.1. Normative References . . . . .	13
6.2. Informative References . . . . .	13
Authors' Addresses . . . . .	14

## 1. Introduction

The ANIMA autonomic management framework addresses the growing management complexity of the highly decentralized and dynamic environment of service provider networks. The ANIMA autonomic management framework will help to produce the unification, governance, and "plug and play" for autonomic networking solutions within existing and future management ecosystems. Three main functional blocks namely the Governance, Coordination and Knowledge functionalities are essential to ensure a proper management and interworking of Autonomic Service Agents (ASAs). This document describes a solution to manage the exchange and processing of information and knowledge between autonomic functions. The objective is to provide a unified interface to enable an interoperable management of information flows among autonomic functions by insuring the use common mechanisms. The protocol negotiate and automatically adapt to the communication and information capabilities, requirements and constraints of the participating entities. The Knowledge functionality plays the role of information / knowledge collection, aggregation, storage/registry, knowledge production and distribution across all the ANIMA functional components (i.e. ANI and ASAs).

The Knowledge block is composed of the following functions:

- o Information Collection and Dissemination - ICD
- o Information Storage and Indexing - ISI
- o Information Processing and Knowledge Production - IPKP
- o Information Flow Establishment and Optimization - IFEO

The Knowledge Block offers basic information/knowledge manipulation functionalities to the ANIMA entities through the Knowledge Exchange Interface. A second interface, the Knowledge Management Interface, handles information flow management that includes configuration actions towards the optimal handling of the information/knowledge in the management system.

## 2. Knowledge Exchange Interface

An Autonomic Service Agent needs two different types of interfaces to deal with the exchange of knowledge.

Knowledge Exchange Interface: Interfaces through which the information are actually exchanged.

Knowledge Management Interface: Interfaces through which the information flows are negotiated, and information capacities are being discovered/advertised. This interface provides configuration actions towards the optimal handling of the information/knowledge in the ASA.

The most important concept is the knowledge exchange flow, which is being set between two knowledge exchange interfaces. It is determined by the two endpoints of the flow and by the type of information that is being conveyed over the flow. Some additional parameters define the way the information are being exchanged (Push or Pull mode plus additional parameters to determine the frequency and conditions of the actual information exchange).

The features of the knowledge exchange flow are being negotiated by Knowledge Management Interfaces and possibly a third party in charge of optimizing the information flows over the whole system. The objective of this negotiation is to determine the characteristics of the exchange flow, which will then be enforced between two/multiple knowledge exchange interfaces.

## 2.1. Information Collection and Dissemination - ICD

The Information Collection and Dissemination (ICD) function is responsible for information collection, sharing, retrieval and dissemination. The ASAs can act as sources or sinks of information. The sources subscribe to the Information Catalog by exposing the type of information they can produce. On the one hand, each information source should subscribe information availability and the equivalent collection constraints (e.g., the supported granularity of collection). On the other hand, each information sink should subscribe information retrieval requirements with a similar process. The subscription process takes place during the ASA bootstrapping. The matching of constraints with requirements takes place during an equivalent negotiation process.

Information can be directly retrieved from or shared with a dedicated Knowledge Sharing system (a sort of ASA which role is limited to be used as a store and sharing entity at the service of other ASAs). As an information collection process is triggered by a component requesting the information, a catalog of the available information has to be built and kept. This catalog indexes which ASA can produce which information. Then upon a bootstrapping ASA requesting a given information to work, the entity in charge of this catalog would then inform requesting ASA of the source ASA. This process could be supported by GRASP discovery mechanism.

The information collection process may be optimized by the Information Flow Establishment and Optimization - IFEO or another utility ASA in charge of optimizing the flows. This ASA acts as the third party during the negotiation phase between an information source and an information sink. If many information sink need the same information, the negotiation entity, is liable to enforce the use of an intermediate Knowledge Sharing system that would collect the information from the source before flooding to sinks according to their requirements.

The collected information may either be directed to the Information Processing and Knowledge Production function for a further processing (e.g., aggregation or knowledge production) and then optionally stored/indexed to the Information Storage and Indexing - ISI function. The storage option may be provided or demanded based on the nature of the information, ASA demands, optimization goals, etc. After this stage, the information or produced knowledge could be passed back to the ICD function for dissemination.

## 2.2. Information Storage and Indexing - ISI

The Information Storage and Indexing (ISI) function is a logical construct representing a distributed repository for registering ASAs, indexing (and optionally storing) information/knowledge. The ISI function stores information, such as ASA registration information and knowledge. The ISI functionality includes methods and functions for keeping track of information sources, including information registration and naming, constraints of information sources, information directory and indexing. An important storage aspect, which can assist the knowledge production handled by the Information Processing and Knowledge Production function, is the inherent support of historical capabilities. For example, an ASA could request information and/or knowledge that was stored in the past using an appropriate time stamp. It should be noted that knowledge production functionality is not part of the ISI function, but it supports the storing of knowledge derived due to some earlier calculations. The ISI optionally stores knowledge produced from the Information Processing and Knowledge Production function (for extended-scoped knowledge) or Knowledge Building ASAs (for locally-scoped knowledge). The different ANIMA entities either requesting or storing information to the Knowledge block, do not directly communicate with the ISI. The ICD function handles information collection or dissemination between the storage points and the ASAs. Furthermore, ISI supports: (i) publish/subscribe information dissemination capabilities, (ii) alternative storage structures (i.e., centralized versus distributed or hierarchical) and database technologies based on the context, and (iii) information and knowledge caching.

## 2.3. Information Processing and Knowledge Production - IPKP

The Information Processing and Knowledge Production function (IPKP) is responsible for operations related to information processing (i.e., aggregation) and knowledge production. The IPKP provides to ASAs and the ANIMA management functions the necessary tool kit to produce different information abstractions, including processed information and extended-scoped knowledge. The Knowledge Production (KP) operation handles and produces knowledge that may be extended-scoped. The latter type of knowledge is being produced out of aggregated information or locally-scoped knowledge. Locally-scoped knowledge can be built from the Knowledge Building ASAs out of data/information directly collected from the managed entities, i.e., its scope is limited to those entities. In all cases of knowledge production, reasoning and inference mechanisms are required. These mechanisms are based on different techniques depending on the exact problem addressed, the type of inputs used and the type of output that needs to be acquired. Such techniques come from scientific areas like statistics, clustering, reasoning, Fuzzy or machine

learning (including supervised, unsupervised and reinforcement learning techniques). All the above information (e.g., problem addressed, type of inputs / outputs, inference/reasoning mechanisms etc) must be described in a proper ontology, ready to be looked up from the IPKP function when such a demand appears. An ASA or ANIMA management function that requires the IPKP functionalities requests to utilize either an Information Aggregation (IA) or a Knowledge Production (KP) operation. The ICD function handles the communication of the ANIMA management component with the internal IPKP functionalities and the IPKP controller is responsible to control the internal IPKP components. The two IPKP operations (i.e., information aggregation and knowledge production) require a number of basic steps:

Step 1: Determining the information aggregation or knowledge production parameters (e.g., information filtering configuration, the inference/reasoning algorithm to use, translation requirements, whether aggregation is required and/or information/knowledge post-processing requirements). This process is being handled from the IPKP controller, which matches the ANIMA component's requirements and the type of problem to solve with the relevant information. The parameters are being communicated to all relevant internal IPKP components.

Step 2: Collection of input information either from an ANIMA component that produces it or from the ISI function (i.e., the knowledge storage). A collection request is being passed back from the IPKP controller to the ICD function.

Step 3: Pre-processing of the input information (e.g., applying information filtering) that may be required. The pre-processing requirements are being set from the IPKP controller.

Step 4: The input information is being passed to the IA operation in case of information aggregation, where an aggregation process takes place according the requirements (e.g., aggregation function used) being set from the IPKP controller. In case of knowledge production, this step may be bypassed or not (i.e., the higher-level knowledge production processes may require aggregation before the inference/reasoning process).

Step 5: In case of knowledge production, the input information may need to be translated in a convenient representation, e.g., to OWL. The translation configuration is being set from the IPKP controller to match the requirements of the inference/reasoning mechanism identified from the (TBD) ANIMA ontology.

Step 6: The actual inference/reasoning process takes place in this step. The input information (i.e., in an appropriate form) and the relevant knowledge production rules are being passed to the identified inference/reasoning mechanism. A rule description language that can be used is the Semantic Web Rule Language (SWRL). The output of this process is the produced Knowledge. This step may be bypassed, in case of a request for information aggregation without knowledge production.

Step 7: The produced knowledge or aggregated information may need a post-processing (e.g., filtering). This step is optional.

Step 8: At this stage, the result is being communicated to the ICD function, to find its way to the requesting ANIMA component. The produced knowledge or aggregated information can be optionally stored in the ISI function so as to be available for ANIMA management mechanisms or ASAs when requested/needed.

#### 2.4. Information Flow Establishment and Optimization - IFEO

The information flow negotiation and optimization aspects are crucial processes overseen from the Information Flow Establishment and Optimization (IFEO) function. The IFEO function, besides organizing internal optimization aspects (e.g., setting filtering or information accuracy objectives), also regulates the information flow based on the current state and the locations of the participating ANIMA components (e.g., the ASAs producing or requiring information). All relevant communication between the knowledge functions and the ANIMA components takes place through the Knowledge Management interface, unless it is otherwise stated.

For clarity purposes, we define the specifications of the IFEO function along with a representative example. We assume the following two ASAs: (a) the Virtual Infrastructure Management (VIM) ASA that provides management and control facilities for virtual infrastructures, including support of traffic monitoring; and (b) the Placement Optimization (PO) ASA that optimizes the data flow over a virtual network through adapting the positioning of communicating nodes (e.g., data servers) in response to the dynamic network conditions. In this example, the VIM ASA provides traffic monitoring information from a particular virtual network to the PO ASA. The PO ASA takes optimization decisions for the network based on this information, i.e., repositions communicating nodes in order to optimize network communication. The information flow negotiation and optimization processes include a number of basic phases, elaborated below:

Phase 1 - Registration: In this phase the ASAs, as part of their registration process with the knowledge block (i.e., described in section 3.6.2), will communicate the following information to the knowledge:

Information they can offer instantly or after an information collection process.

Knowledge they can offer instantly or after a knowledge production process.

Information/knowledge they would require (mandatory or optional).

The above information is embedded in the description of the ASA instance description. In our example, the VIM ASA registers the information it can offer (e.g., the topology information and measurements on the link loads). This information can be offered instantly (i.e., does not require an information collection process to start, since it monitors the network continuously). The PO ASA registers the same information type as mandatory information required.

Phase 2 - An ANIMA management function requesting knowledge: In this phase a process in an ANIMA management function (like a supervision process in management or a knowledge production mechanism or a coordination mechanism) demands to register to a given piece of information produced by a given ASA. This information is expressed as a information specification. In that case, the Knowledge Management Interface of the requesting entity is calling a TBD knowledge method named to request the registered information.

Phase 3 - Information Flow Negotiation: In the third phase, the knowledge block through its IFEO function handles a flow negotiation process between entities (i.e., ASAs or management mechanisms) requiring information and those can provide it. The two entities exchange information flow related parameters with the knowledge block, in order to confirm that all information-related requirements can be satisfied under the given constraints. An information flow is either established between the two entities directly or between an entity and the knowledge block itself, in case the requested information is available in the knowledge storage. The negotiation process includes flow-level optimization aspects as well. This phase is composed of the following steps:

Step 1 - Preselecting the information flow ends: Whenever a ASA registers it advertises requested information/knowledge (under

a specific format TBD), the knowledge block fetches in its indexing storage the appropriate entity (ASA or management mechanism) that can produce the requested information/knowledge. It may either select an entity by considering the type of information/knowledge required or, in case of alternative options, assign the first entity it finds and enlist the other potential choices in a queue. In case the required information is in the knowledge storage, an information flow is created with the knowledge. The same process happens when a ANIMA management entity requests some knowledge, depending on the form of the request (i.e., a fetching from the indexing storage may or may not be required):

ASA information: already specifies which is the Instance ID of the ASA producing the information.

ANIMA information: a fetching from the index table is required to pick the appropriate flow ends.

Management information: then the fetching does not concern finding a flow end, but finding all the flow ends matching the pattern provided by the management information in order to establish as many flows as indexed ANIMA information objects inheriting from the management information (this corresponds to a supervision mechanism requesting to register to ASA utilities, hence a flow for each ASA capable of advertising its utility will be created). Reversely, knowledge may have postponed flow establishments of some requested information because at the time the request was received, no entity producing this information was registered. In that case, knowledge checks with every received instance description whether the advertised information matches previously unsolved requests. After that, the IFEO proceeds to Step 2. In the studied example, the knowledge block preselects the VIM ASA as information source for the PO ASA that acts as the information sink. This selection was based on the matching information URIs referenced in the registered ANIMA information data structures from the two ASAs.

Step 2 -Communicating the negotiation parameters: in step 2, a negotiation process is initiated between the entity requiring information/knowledge (i.e., the information sink entity) and the selected information source entity. The negotiation begins with the two entities communicating additional negotiation parameters to the knowledge block. Specifically, the information sink entity communicates an augmented version of the ANIMA information with:

- QoS Requirements on the information/knowledge it requires.
- Preferred information communication method (i.e., either push/pull or pub/sub).
- List of Knowledge Exchange interfaces (addresses) on which the information can be received and possibly an internal metric regarding the internal costs to use this information from each of these interfaces.
- REST callback functions that may be required at this end of communication (e.g., in case of an information subscribe method).

In a similar way, the information source entity communicates the following to the knowledge block:

- QoS Constraints on the information/knowledge it can offer.
- Supported (and preferred) information communication method (i.e., either push/pull or pub/sub).
- Whether for this requested information/knowledge an "information collection/knowledge production" process is already activated or needs to be initiated.
- List of Knowledge Exchange interfaces (addresses) on which the information can be provided and possibly an internal metric regarding their internal cost to bring this information up to the interface.
- REST Callback functions for the relevant capabilities (i.e., triggering functions for information collection or knowledge production - if relevant).

Practically, the knowledge block initiates a new negotiation with the execution of the sink and source parameters negotiation methods of the Knowledge Management Interface. Both methods take as input the specifications of the information to be communicated from the established communication flow, represented as an ANIMA information data structure. In the reference scenario, the VIM ASA communicates to the knowledge: (i) the QoS constraints of the topology and link load information it can offer, e.g., monitors information once per 10 secs, and (ii) a number of available Knowledge Exchange interfaces that can provide the information. The PO ASA communicates to the knowledge: (i) the QoS requirements of the required information, e.g., once per 30 secs, and (ii) a

number of available Knowledge Exchange interfaces that can receive the information.

Step 3 - Completing the negotiation: The knowledge block matches information flow requirements with constraints, determines the information flow parameters with flow optimization considerations and then issues a Knowledge Exchange Policy summarizing an information flow contract to both entities. knowledge also stores the Knowledge Exchange Policy through the Information Flow Configuration and Statistics operation of the IFEO function. In case of an unsuccessful negotiation (i.e., the requirements do not match the constraints), it may disengage or trigger a new negotiation:

- a) With the same information source entity but lower requirements.
- >b) With another information source entity that waits in the queue, until the queue is exhausted.

The Information Exchange Policies for the corresponding flow are being produced from the Information Quality Controller operation of the IFEO knowledge block function and include:

- Location/addresses of the participating Knowledge Exchange Interfaces in the information flow.
- Internal knowledge optimization decisions that may impact the information flow (e.g., optimal knowledge aggregation/storage points), in case the knowledge block is the one end of the flow.
- Addresses of triggering callback functions for knowledge production or information collection - if relevant.

These policies are considering the requirements/constraints of the participating entities and the global performance objective coming from the operator (e.g. via the ANIMA Intent Policy). The knowledge establishes the information flow using a set flow method of the Knowledge Management Interface, that takes as an input the decided Information Flow Exchange Policies, represented as a flow data structure.

The decided Information Exchange Policies are being applied to the network through the respective ASAs or communicated to the knowledge functions they are associated with. Since the appropriate context environment for the new information flow is prepared, a suitable path

between the participating nodes is established next. This process considers the locations of the entities producing and requiring information and the required knowledge nodes (e.g., aggregation points, storage points etc) as well as the potential traffic characteristics. After that, the Knowledge Exchange interface can be accessed anytime from the information sink entity to receive the needed information/knowledge. In our reference scenario, the knowledge block matches the information flow constraints (e.g., supported monitoring rate) of the VIM ASA with the information flow requirements from the PO ASA. Then it selects the most appropriate Knowledge Exchange interfaces to communicate the information from the VIM to the PO ASA. A new information flow contract is established and communicated to the two ASAs and stored in the knowledge block. The information flow is established and the PO ASA can retrieve the required information from the VIM ASA via the appropriate Knowledge Exchange interface. The PO ASA can now begin taking network optimization decisions using that information.

Knowledge-level Optimization: Furthermore, knowledge supports a global optimization process that is triggered periodically or when a global performance objective change is requested from the GOV. This process takes optimization decisions using the aggregated information from the configuration of all established information flows and is related with a restructuring of the knowledge functions themselves. In other words, global-optimization algorithms may discard or update Knowledge Exchange Policies enforced for established information flows. It takes as an input the global picture of all the established information flow contacts and provides as an output different contracts aligned better to the new updated demands (e.g., a new received global objective). This process may initiate re-negotiations that include requesting again from the entities what their requirements and capabilities are. For example, the distributed knowledge nodes may be increased, decreased or repositioned in order to accommodate all established information flows and the global optimization goal better. The optimization process is triggered by the IFEO function and regulates the information flow based on the current state and the locations of the participating ANIMA components. In particular, the IFEO controls information collection handled from the ICD function, information aggregation, and aggregation node placement. Furthermore, it guides a filtering system for information collection and aggregation points that can significantly reduce the communication overhead.

### 3. Acknowledgments

This draft was written using the xml2rfc project.

The content of this draft builds upon work achieved during the EU FP7 UniverSelf project ([www.univerself-project.eu](http://www.univerself-project.eu)).

#### 4. IANA Considerations

This memo includes no request to IANA.

#### 5. Security Considerations

TBD

#### 6. References

##### 6.1. Normative References

[I-D.ciavaglia-anima-coordination]

Ciavaglia, L. and P. Peloso, "Autonomic Functions Coordination", draft-ciavaglia-anima-coordination-00 (work in progress), July 2015.

[I-D.peloso-anima-autonomic-function]

Peloso, P. and L. Ciavaglia, "A Day in the Life of an Autonomic Function", draft-peloso-anima-autonomic-function-00 (work in progress), October 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

##### 6.2. Informative References

[I-D.behringer-anima-reference-model]

Behringer, M., Carpenter, B., Eckert, T., Ciavaglia, L., Liu, B., Jeff, J., and J. Strassner, "A Reference Model for Autonomic Networking", draft-behringer-anima-reference-model-04 (work in progress), October 2015.

[RFC7575] Behringer, M., Pritikin, M., Bjarnason, S., Clemm, A., Carpenter, B., Jiang, S., and L. Ciavaglia, "Autonomic Networking: Definitions and Design Goals", RFC 7575, DOI 10.17487/RFC7575, June 2015, <<http://www.rfc-editor.org/info/rfc7575>>.

Authors' Addresses

Laurent Ciavaglia  
Nokia  
Villarceaux  
Nozay 91460  
FR

Email: [laurent.ciavaglia@nokia.com](mailto:laurent.ciavaglia@nokia.com)

Pierre Peloso  
Nokia  
Villarceaux  
Nozay 91460  
FR

Email: [pierre.peloso@nokia.com](mailto:pierre.peloso@nokia.com)