Internet-Draft
Intended status: Standards Track
Expires: December 2, 2016

Y. Xia, Ed.
T. Zhou, Ed.
Y. Zhang, Ed.
S. Hares
Huawei
P. Aranda
D. Lopez
Telefornica
J. Crowcroft
Cambridge University
Y. Zhang
China Unicom
May 31, 2016

Intent Common Information Model
draft-xia-ibnemo-icim-02

Abstract

Intent Common Information Model (ICIM) defines a unified model for expressing different layers' intent whatever role, responsibility, knowledge, etc. This document provides an information model to be inherited and expanded to construct specific intent model in different areas. According to this information model, network intent model is put forward which can satisfy users' need in different layers, such as, end-users, business developers, and network administers.

Copyright Notice

Table of Contents

1. Introduction

Network operations have traditionally been designed bottom-up starting with low level device interfaces designed by protocol experts.In order that interfaces could be wildly used by various users, information details are exposed as much as possible. It enables better control of devices, but leaves huge burden of selecting useful information to users without well training. Since the north bound interface (NBI) is used by network users, a more appropriate design is to express user intent and abstract the network

from the top down. The intent base NBI expresses what a network service consumer (e.g., application, operator) requires from the network but it does not completely specify or constrain how that service may or should be delivered by the network. The intent is expected to be independent of protocols, network interface styles, vendor features, media attributes, or any other network implementations.

Intent Common Information Model (ICIM) specifies a generic model for expressing key components of intent interface and the relationship between these components. This document provides a common model which could be inherited from and expanded to construct specific intent interface in dedicated areas. According to this information model, intent interface in network area can satisfy users' intention in different roles, such as, end-users, business developers, network administers, etc.

1.1. Terminology

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].

2. Intent Common Information Model Overview

Intent Common Information Model aims to specify a unified information model which satisfied different areas, scenarios, and other constraints. So, it is a complete and detailed information model to define the constituent elements of intent. However, not all elements need to be present when mapping this model to a specific data model, since some of the elements can be obtained by system automatically. From the overall perspective, construction elements of intent can be described as:

-user of intent who author and own this intent

-intent content which is a desired purpose and

-the specific context which is the background circumstance information.

Furthermore, in general, person's intent content usually describes the ultimate state of some objects or applies actions to these objects. So intent content can be abstracted into further:

-object which is the target for intent

-result which is a desired state and

-operation which is the specific actions to achieve a purpose.

2.1. Elements

2.1.1 Users

User is an abstract class which specifies the subject and owner to express the intent. It is a performance of roles in real world, that is, each user serves as a role or a combination of roles actually. For example, end-users, business designers, network administrators are all instances of User class which act as specific roles. When a user is labeled as a role, he will have the desire and requirement to express intent belonged to this role. Owning to different network abstraction views, intent is different for specific user when this information model is applied to specific scenario.

Though one user serves as one role in most cases, it is sensible and acceptable that one user serves as multiple roles and intent of these users may involve more functions and huge operation scope.

2.1.2 Context

Context is an abstraction class which refers to a set of specific background information such as, timer, price, and so on. Context has a huge influence on a person designing a detailed plan or selecting the best program to achieve a purpose. For example, when an enterprise plans to build a dedicated connection between two sites, price and distance will be the context in this scenario. While may not be part of how an entity expresses or executes some intent, it is a factor that must be considered with the expression of intent.

2.1.3 Object

Object is an abstract class which refers an abstract class which defines some entities affected or managed by intent. For the management, users could manage life cycle of the objects through some concrete operations, such as, create, update, delete, etc. In addition, users could use other specific operations to affect the behavior of managed objects. For example, a business designer want all traffic be filtered by a special firewall. The object of this business designers intent could be the all traffic flowing on a specific network (e.g. L3VPN), and this intent impacts the forwarding behavior of the traffic network. Object is different in specific area. In network area, object is an aggregation class with CustomerFacingNode, Connection and ServiceFlow. For objects, users could construct some specific objects to achieve intent, and it is also allowed for users to assign intent to existing resources which is physical/virtual devices or defined by other ways.

2.1.4 Result

Result is a type of intent which refers to an final state or something an individual wants to achieve. This type of intent shields difference and diversity of an environment away from the users' intent. The person just describes the final state of objects without worrying about how to achieve it. For example, a result could be that the company accesses any sites on the Internet safely. It just defines a result that ignores technology details, such as, firewall, ACL, and so on.

In addition to the expecting state, violation is another special state which has an important status when achieving integral compliance. For example, a typical scenario is that one specific tenant does not want his virtual machines to share a some hypervisor with other tenants. This type of result just shows the undesired state which express users' intent, so this kind of intent should be another type of result.

2.1.5 Operation

Operation is a type of intent which refers to some specific actions an individual desires to take for realizing the purpose. This type of intent formulates explicit plan to realize a purpose which may take a better control of the whole system. According to the diversity of system support capability, there are large sets of operations for users to take.

Generally, operations can be divided into two categories. One is action without condition. For example, create a virtual machine. This kind of operation defines a concrete action which is executed immediately without any trigger. The other is action with condition. For this kind of operations, condition is a trigger for the action. And actions will not be executed immediately until the condition clause is tested to be true. For example, "do load balancing when the utilization of a link exceeds 80%". In this example, "utilization of a link > 80%" is the trigger, and "do load balancing" is the action. Action will not be executed until the trigger is true. Actions are different according to users' role which has different abstraction views. And actions will not be detailed configurations in devices, but high-level and packaged functions which are translated into configurations. For example, the service providers' action "do load balancing" is device independent, and network operators' action may configure load balance pools depending on specific devices.

2.2. Relationships in ICIM

2.2.1 Relationship between Result and Operation

Users are free to express their intent, no matter it is an final result or specific operations in their mind, but there are some relationships between these two basic types of intent. Result refers to an ultimate and relatively permanent status, regardless which ways to maintain it. However, operations specify what kinds of action need to take explicitly, which more focus on temporary or specific behavior to achieve some goal. One typical service scenario is that all links' utilization should not exceed 80%. By way of Result, the intent will be expecting all links' utilizations are smaller than 80% (or avoiding any link' utilization exceeds 80%). By way of Operation, the intent will be if links' utilization exceeds 80%, redirect some flows to other links (or some other actions could achieve this goal).

For result, users just need to express the goal without worrying how to implement it in a specific system which allows users to focus on real requirement. To achieve the result, it needs some reasoning mechanisms to transfer it to real executable operations which are supported by specific system. So in a specific scenario, a result can generate a set of concrete operations. For the above example, if user just expresses the result, that is, all links' utilizations are smaller than 80%. The system will choose suitable operations to achieve this status automatically, i.e., expand the capacity of links whose utilization exceed 80%, or redirect flows to other links whose capacities are far less than 80%.

2.2.2 Relationship between Object and Operation

Operation refers to some specific actions on some objects, so object is the target of an action. In general, any action will include some objects to execute this action. When users want to execute some actions to achieve goals, they may construct the target objects and assign specific actions on them, and it is allowed for users to use existing resources to do some operations. Though object is the target of action, it offers the constraint for optional operations. For example, for a virtual machine, the optional operations are create, delete, migrate, etc.

2.2.3 Relationship between Object and Result

Result refers to some final state for some objects. This type of intent does not define which specific operations to take, but only express the desired state of objects. So it is independent on objects' concrete capability. For example, intent is all virtual

machines' CPU utilization could not exceed 80%. It does not assign specific operations. So reasoning mechanism will choose suitable operations to satisfy this intent, such as, migrate virtual machine or expand it.

2.3. Intent and Policy

In industry, Policy already has a clear definition, such as in RFC3060. Policy rule consists of an event, a set of conditions and a set of actions. When an event occurs, actions will be taken until condition clauses are evaluated to be true.

As mentioned above, intent refers to a purpose in achieving result or performing operation. The intent has a larger scope compared with the policy since Intent can express both result and operation. On one hand if a result is described by intent, there may be no specific action given to show how to achieve this intent. On the other hand, if operation described by intent, conditions of action is optional. Policy is a specific form of operation in intent.

2.4. Role-based Intent

In an integrated system, roles are divided into several categories according to the division of work, architecture of system, etc. In network system, network abstraction will be quite different in the perspective of each role. So intent has strong dependencies on roles. Intent expressed by different categories of roles will focus on different points and have different intent expression.

For example, if an agent is labeled as service provider role, he may just care about the high-level services, such as, security communication. And if he is labeled as network architecture role, he will care about the details of the whole architecture.

2.5. Intent and FSM

Intent, standing on the perspective of users, expresses intuitive service requirement, including desired network services or means to fulfill network goals. In other words, the Intent model defines a series of FSMs(finite-state machines) and transition between these state machines to implement the managment of service's life cycle.

Specifically, each state machine, combining with elements which construct intent, represents a specific state of one or several objects, for example, a normal-work state or exception state of a connectivity service. Both current state(or initial state) and result state(or desired state) are one type of state machines, the main difference between which is that result state is an ultimate

state repects users' requirement, otherwise, current state represents a initial state needs to be transited to others states to satisfy users' intent.

Operation, including specific actions, represents the transition between current state and result state that describe the state of objects as mentioned above. The transition between service states could be shown as below.

```
                        operation
                +-------------------------+
                |                         |
         +------+------+           +------+-----+
         |current state+-----------+result state|
         +-------------+           +------------+
```

Figure 1 intent and FSMs

Just as mentioned in the section 2.2, operation is the mean to change network service state to fulfill users' intent, namely, the result state. For users, who clearly grasp how to reach result state or have to assgin actions for other reasons, they could descibe specific operations in intent to realize the result. Otherwise, users just need to descibe what's the result or desired state, and the complier system will produces specific operations to achive the result.

A typical scenario is the constraint for bandwidth utilization. For user who does not know the ways to adjust bandwidth, he may express his intent with desired state, namely, the result state is bandwidth-utilization<80% and system will choose suitable ways to adjust bandwidth, such as, expansion or changing route. But for users who has experience with network operation, he may express his intent with operaiton, namely, do load balancing if bandwidth-utilization>80%. Both expressiones descibe service state machines and transition, but operation could implement the transition between current state and result state.

3. Intent Modeling

This section defines the concept and hierarchy of intent, and describes the Intent Common Information Model.

3.1. Notation

The notation used in this document is adapted from the UML (United Modeling Language). We will use the UML for the intent information modeling. This section listed symbols that will be used in this document for relationships among information models.

-->

Stands for the association relationship. Association represents the static relationship shared among the objects of two classes.

--A

Stands for the aggregation relationship. Aggregation is a variant of the "has a" association relationship; aggregation is more specific than association. It is an association that represents a part-whole or part-of relationship. Aggregation can occur when a class is a collection or container of other classes, but the contained classes do not have a strong lifecycle dependency on the container. The contents of the container are not automatically destroyed when the container is.

--C

Stands for the composition relationship. Composition is a stronger variant of the "has a" association relationship. It is more specific than aggregation. Composition usually has a strong lifecycle dependency between instances of the container class and instances of the contained class. If the container is destroyed, normally every instance that it contains is destroyed as well.

--G

Stands for the generalization relationship. The Generalization indicates that one of the two related classes (the subclass) is considered to be a specialized form of the other (the super type) and the super class is considered a 'Generalization' of the subclass. In practice, this means that any instance of the subtype is also an instance of the super class.

3.2. Intent overview

In general, intent is one's specific mental activity, so it strongly depends on the subject. Different users may have different intent. In addition, context, omitted usually, is an important factor when achieving purpose, which offers necessary background information to impact the decision. It illustrates the overview of the intent. Figure 2 indicates that the user has intent in some context. For example, an enterprise wants to block all http traffic in work time. In this intent, the user is the enterprise, the intent is to block all http traffic in the work hours, and the context includes the definition of the "enterprise" and the "work hours".

```
                   +------+ has +------+ in  +-------+
                   | user +---->+intent+---->+context|
                   +------+     +------+     +-------+
                  Figure 2 general prescription for intent
```

3.3. Top level intent expression

In Cambridge Dictionaries, the definition of "intent" is the fact that you want and plan to do something. So, in general, intent refers to an agent's purpose on getting the result or performing some specific operation. In specific areas, these results or operations will relate to some objects. Figure 3 describes the general expression of intent.

```
                                +--------+
                                | intent |
                                +-C-A-A--+
                                  | | |
                      +-----------+ | +-------------+
                      |             |               |
                      |             |               |
                  +---+---+      +--+---+     +-----+-----+
                  | object|      |result|     | operation |
                  +-------+      +------+     +-----------+
                        Figure 3 intent expression
```

One type of intent is to express key operations that a user wants to execute. The underlying intent system can generate a complete operation list from user's request. The other type of intent is to express the result or state without dictating any operations.

For example, intent of a user may be a result without defining how to realize it, such as, requiring security communication between two sites, or dictate some detailed operations in order to achieve a purpose, such as, filtering all traffics by firewall between these two sites.

3.4. Objects in the network

Object is an abstraction class which can be inherited from and expanded in different area. It, cared about by users, represents the target of result and operations. In network area, the object, i.e. the target of intent, can be generalized into CustomerFacingNode(CFN), Connection and ServiceFlow, as shown in Figure 4.

```
                                +-------------------+
                            +---+CustomerFacingNode|
                            |   +-------------------+
                +-------+    |
                |      +G+--+   +-----------+
                |Object+G+------+Connection|
                |      +G+--+   +-----------+
                +-------+    |
                            |   +------------+
                            +---+ServiceFlow|
                                +------------+
```

Figure 4 common objects in network area

The CustomerFacingNode represents the functions a user-facing network node may provide in a network such as network services, forwarding functions (firewall, load balancer, virtual router, and others), or a group of network elements. A group of network elements can be a subnet, an autonomous system, or a confederation of autonomous systems.

The Connection describes the link resources between two CFNs. These link resources construct the foundation of communications between different CFNs. User could take connection as physical link, and assign bandwidth on it.

The ServiceFlow refers to the traffic in network which describes data packets have some certain common characters. ServiceFlow model describes the connectivity in virtual network, namely, if users want to describe the communications between CFNs without direct connection, they have to define the service flow and assign operation to allow the service flow.

3.5. Type of result

Result refers to the final state which is expected or avoided. Figure 5 describes two types of result. Both of the results just show the performance of some objects, without caring about how to reach them.

Result could be expressed as a set of logic clause connected with propositional literals including AND, OR and NOT. The logic clause could be described as an expression with relational operators, such as equal, greater-than, less-than, belong-to.

With this model, users could express the desired state as an expression. System will resolve the expression and seek ways to make it true. The result will be achieved when the expression is evaluated to be true. The typical examples are shown as follows:

- For example, a user may express an intent as the network link utilization must less than 80%. This expression is a type of result which describes an expected state. The left operand is the utilization of all links, the right operand is 80%, and the operator is less-than.

- Another example is an enterprise wants the development team and sales team not to share a common link. In this intent, the left operand is the union of the link set of development team occupied and the link set of sales team occupied. The operation will be equal, and the right operator is an empty set.

Though a unified information model for the Result is proposed in here, it is still a preliminary version which just expresses the basic components. The formalization and standardization are still open issues need to be studied further. More comprehensive and detailed manifestations will be added in the next version.

```
                  +--------+
                  | result |
                  +--G-G---+
                     | |
               +-----+ +-----+
               |             |
           +---+----+    +---+---+
           | expect |    | avoid |
           +--------+    +-------+
```

Figure 5 expression of Result

3.6. Operation composition

Operation refers to some specific actions in order to achieve some purposes. An operation must have some actions. However, if condition and constraint can be optionally defined in operations, it depends on specific scenario and users' requirement. Once a condition is involved in operation, actions will not be executed immediately until condition is true. In additional, constraint restricts action itself or the scope of action.

For example, redirect traffic to back-up link when the utilization of current link exceeds 80%, except the flow from VIP user. In this scenario, the condition is link utilization exceeds 80%, the action is redirect traffic, and the constraint is VIP flow could not be redirected.

```
                    +---------+
                    |operation|
                    +--A-C-A--+
                       | | |
             +---------+ | +------------+
             |           |              |
             |           |              |
        +----+----+   +--+---+    +-----+----+
        |condition|   |action|    |constraint|
        +---------+   +------+    +----------+
         Figure 6 composition of operation
```

4. Security Considerations

TBD

5. IANA Considerations

This draft includes no request to IANA.

6. Acknowledgements

The authors would like to thanks the valuable comments made by Wei Cao, Sheng Jiang, Zhigang Ji, Xuewei Wang, Shixing Liu, Yan Zhang.

7. Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Authors' Addresses

Yinben Xia
Huawei Technologies Co., Ltd
Q14, Huawei Campus, No.156 Beiqing Road Hai-Dian District
Beijing 100095
China

Email: xiayinben@huawei.com

Tianran Zhou
Huawei Technologies Co., Ltd
Q14, Huawei Campus, No.156 Beiqing Road Hai-Dian District
Beijing 100095
China

Email: zhoutianran@huawei.com

Yali Zhang
Huawei Technologies Co., Ltd
Q14, Huawei Campus, No.156 Beiqing Road Hai-Dian District
Beijing 100095
China

Email: zhangyali369@huawei.com

Susan Hares
Huawei Technologies Co., Ltd
7453 Hickory Hill Saline
MI 48176
USA

Email: shares@ndzh.com

Pedro Andres Aranda
Telefornica I+D
Don Ramon de la Cruz, 82 Street Madrid
28006
Spain

Email: pedroa.aranda@telefonica.com

Diego R. Lopez
Telefornica I+D
Don Ramon de la Cruz, 82 Street Madrid
28006
Spain

Email: diego.r.lopez@telefonica.com

Jon Crowcroft
University of Cambridge Computer Laboratory
William Gates Building, 15 JJ Thomson Avenue Cambridge
CB3 0FD UK

Email: jon.crowcroft@cl.cam.ac.uk

Yan Zhang
China Unicom P.R.
China

Email: zhangy1036@chinaunicom.cn