A YANG Data model for Event Management

draft-wwx-netmod-event-yang-05

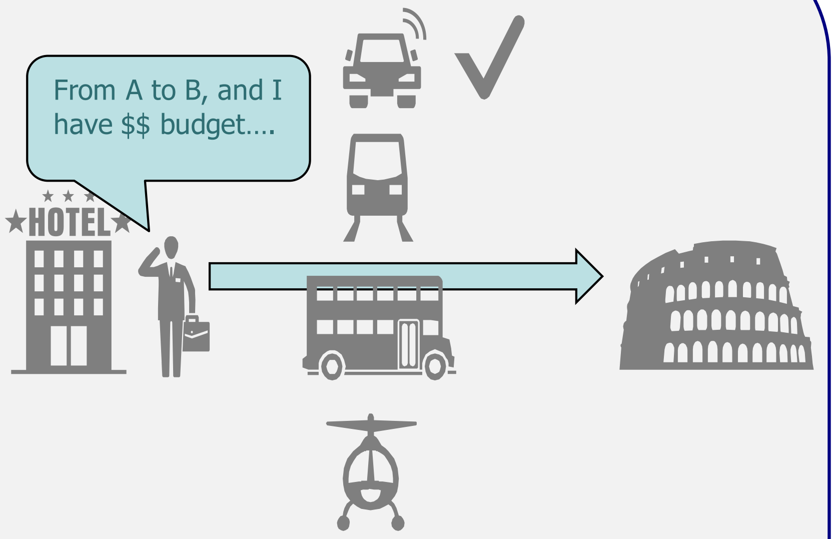
Authors:

- M. Wang (wangzitao@huawei.com)
- Q. Wu (bill.wu@huawei.com)
- C. Xie (xiechf@ctbri.com.cn)
- I. Bryskin (i_bryskin@yahoo.com)
- X. Liu (xufeng.liu.ietf@gmail.com)
- A. Clemm (ludwig@clemm.org)
- H. Birkholz (henk.birkholz@sit.fraunhofer.de)
- T. Zhou (zhoutianran@huawei.com)

Background – What is ECA?

- Policy discussed in RFC8328 are classified into imperative policy and declarative policy, ECA policy is an typical example of imperative policy.
  - Declarative policy : express the goal, the targeting state
  - Imperative policy: how to achieve those goals
- Event-Condition-Action is a shortcut for referring to the structure of active rules in event-driven architecture and active database systems;

- An ECA policy rule is activated when its event clause is true; the condition clause is then evaluated and, if true, signals the execution of one or more actions in the action clause.



Declarative policy : express the goal, the targeting state

Imperative policy: how to achieve those goals

Updates since the IETF 105

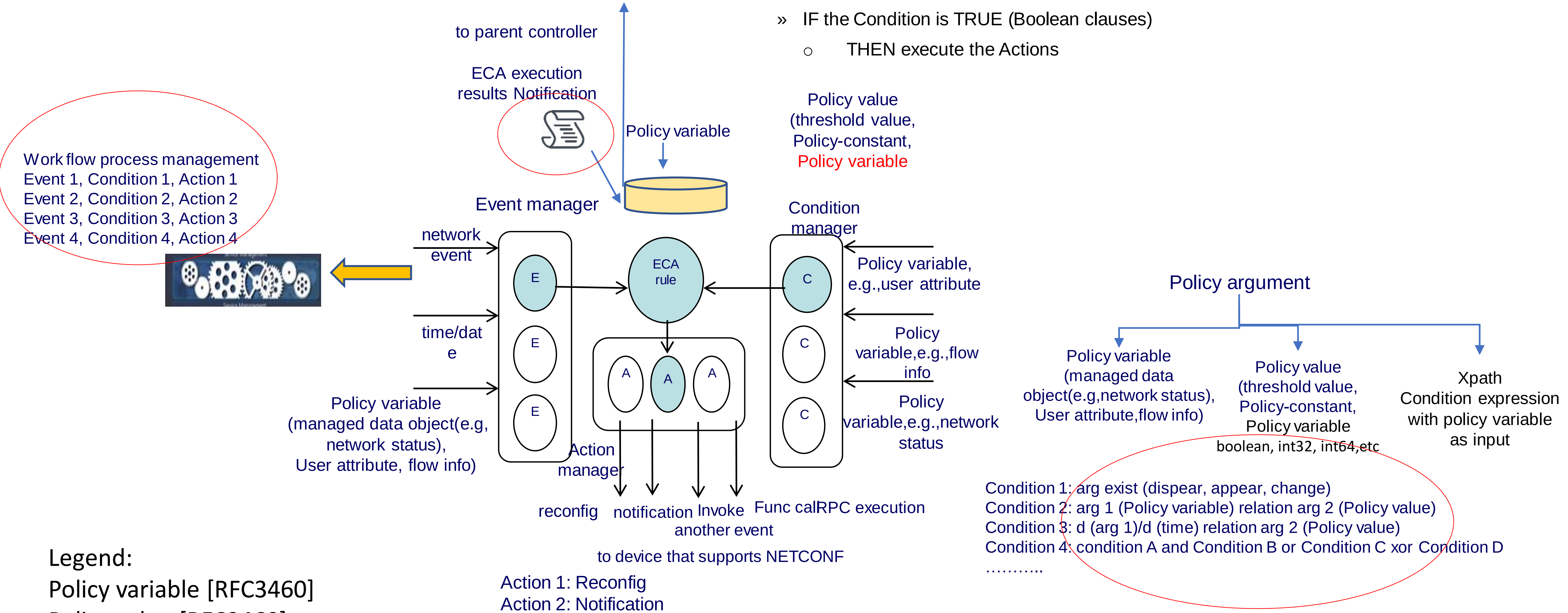
- Presented the -02 version in the last NETMOD session and got a good number of supports when the chair polled in the room.
- Chairs suggest to harmonize with ECA related draft, i.e.
  - draft-bryskin-netconf-automation-yang
- Three updates are issued before this meeting:
- 04-05:
  - Harmonize with draft-bryskin and fold additional attributes in the models (e.g., policy variable, func call enhancement, rpc execution);
  - ECA conditions part harmonization;
  - ECA Event, Condition, Action, Policy Vriable and Value definition;
  - Change ietf-event.yang into ietf-eca.yang and remove ietf-event- trigger.yang
- 03-04:
  - Update objective section to align with use cases.
  - Clarify the relationship between target and policy variable.
  - Change variation trigger condition back into threshold trigger condition and clarify the usage of three trigger conditions.
  - Remove Event MIB related section.
  - Add new coauthors.
- 02-03:
  - Usage Example Update:
    - Add text in introduction section to clarify the usage examples of ECA policy

What have we done

- Per chair’s request, authors of both draft discussed on the list on possibility of coming up unified ECA proposal
  - Commonality:
    - Basic Functionalities (E,C,A,Policy variable) and Use cases
      - Network failure recovery, smart filter
  - Advance functionalities need to be agreed (condition expression, function call, RPC call, etc)
  - Terminology alignment
    - Policy variable vs target
    - Trigger condition vs condition expression
  - Policy variable definition and Purpose
- We met as a team (in Singapore) on Monday morning to decide on how to scope the work (https://ietf.org/how/meetings/106/side-meetings/)
  - Agreement that is in scope is to
    - add various type of policy variable support (e.g., policy variable, explicit policy variable, implicit policy variable(boolean, int32, int64) add condition expression support, func call support, RPC call support,
    - decouple condition and action from Event in the ECA model Framework
    - Focus on network control logic delegation to the device that supports netconf protocol.
  - Agreement that is not in scope is
    - ECA model invokes ECA script,
    - centralized ECA Policy control (Action executed in the upper layer control element)
    - smart filter model (that extends from ECA basic model)

ECA Model Design

- Event-Condition-Action (ECA)
  - E.g. IF the Event is TRUE (Boolean clauses)
    - IF the Condition is TRUE (Boolean clauses)
      - THEN execute the Actions



to parent controller

ECA execution results Notification

Policy variable

Policy value (threshold value, Policy-constant, Policy variable

Work flow process management
Event 1, Condition 1, Action 1
Event 2, Condition 2, Action 2
Event 3, Condition 3, Action 3
Event 4, Condition 4, Action 4

Event manager

Condition manager

network event

time/date

Policy variable (managed data object(e.g, network status), User attribute, flow info)

E
E
E

ECA rule

A
A
A

Action manager

C
C
C

Policy variable, e.g.,user attribute

Policy variable,e.g.,flow info

Policy variable,e.g.,network status

reconfig

notification

Invoke another event

Func calRPC execution

to device that supports NETCONF

Policy argument

Policy variable (managed data object(e.g,network status), User attribute,flow info)

Policy value (threshold value, Policy-constant, Policy variable

boolean, int32, int64,etc

Xpath Condition expression with policy variable as input

Condition 1: arg exist (dispear, appear, change)
Condition 2: arg 1 (Policy variable) relation arg 2 (Policy value)
Condition 3: d (arg 1)/d (time) relation arg 2 (Policy value)
Condition 4: condition A and Condition B or Condition C xor Condition D
………..

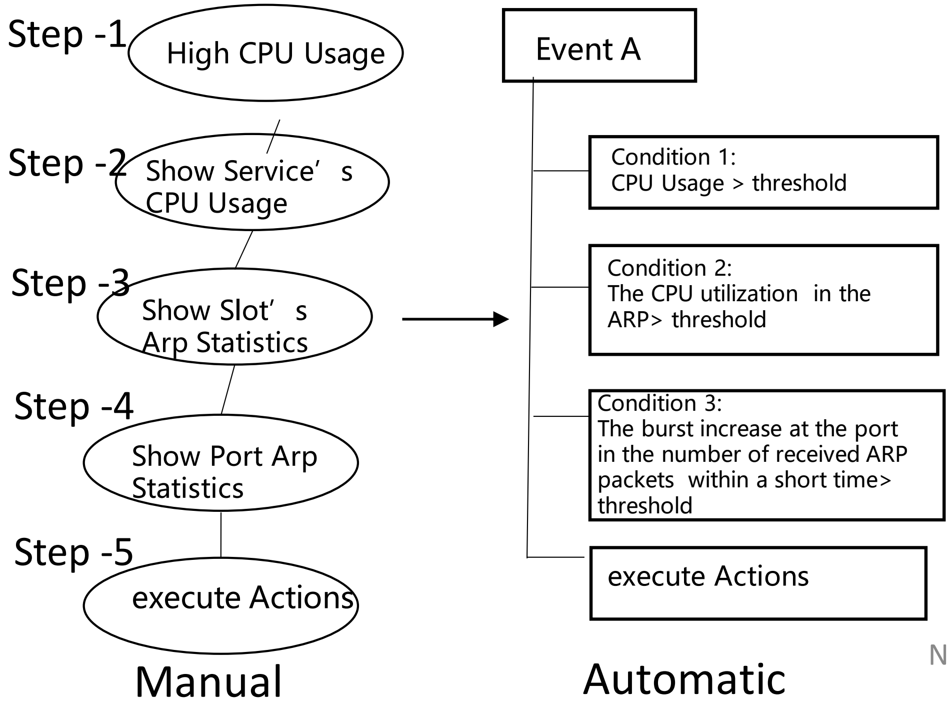
Legend:
Policy variable [RFC3460]
Policy value [RFC3460]
Policy argument
ECA [RFC8328]

Action 1: Reconfig
Action 2: Notification
Action 3: Invoke another event
Action 4: Func Call
Action 5: RPC execute

Targeted Use Cases

Supported Use Cases	Threshold	Threshold exceeding times	Condition expression (AND, OR,XOR)	Stateless or stateful?	Min, max, variance, average, etc, computation intensive	RPC execution support
Fault localization and self-healing	Y	Y	Y	Y	N	N
Telemetry Smart Filter	Y	Y/?	Y	N	N	N
TE path computation	Y	Y	Y	Y	Y	Y

1. Fault localization and self-healing
Example: ARP attack



Step -1 High CPU Usage

Step -2 Show Service' s CPU Usage

Step -3 Show Slot' s Arp Statistics

Step -4 Show Port Arp Statistics

Step -5 execute Actions

Manual

Event A

Condition 1:
CPU Usage > threshold

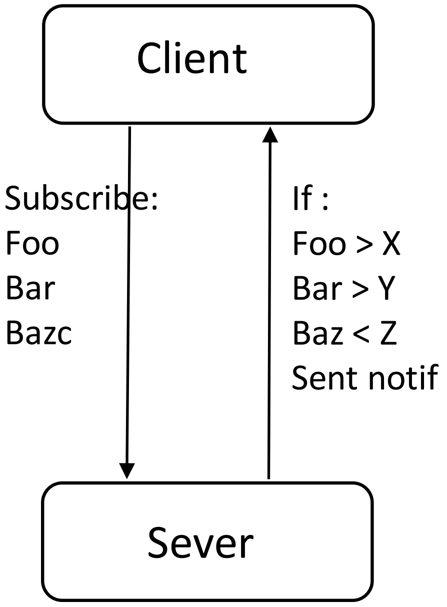
Condition 2:
The CPU utilization in the ARP> threshold

Condition 3:
The burst increase at the port in the number of received ARP packets within a short time> threshold

execute Actions

Automatic

2. Telemetry Smart Filter
Example:



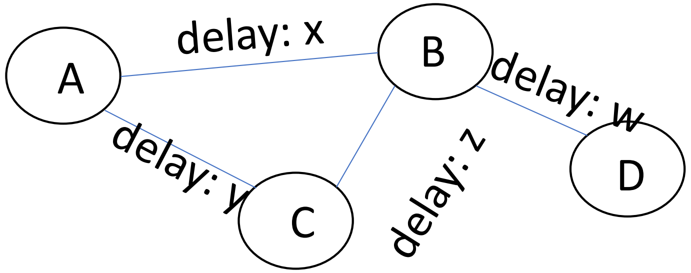
Client

Subscribe:
Foo
Bar
Bazc

If :
Foo > X
Bar > Y
Baz < Z
Sent notif

Sever

3. TE path computation
Example



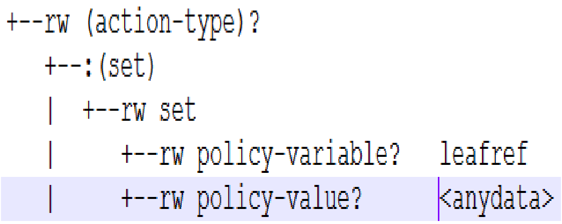
Example of policy: if(service_destination matches 10.132.12.0/24) Use path:
A=> B => D.
else Compute path with minimal delay.

Next Steps

- Two draft authors have agreed to work together.
- Keep on adding clarity to the documented scope and solicit feeback and input.
- Question to chairs: Is this draft a good baseline for the next step?

Proposal: How to use PVs in the ECA Action

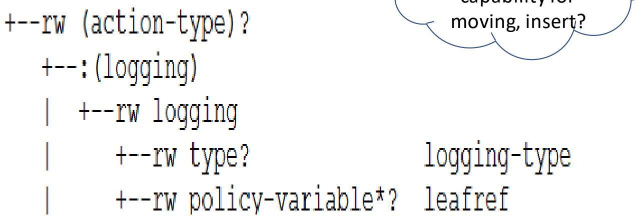
- How the client can use PVs in 1) reconfiguration, 2) notifications sent to the client1 3) computation actions, 4) RPC input/output ?



```
+--rw (action-type)?
   +--:(set)
   |  +--rw set
   |     +--rw policy-variable?   leafref
   |     +--rw policy-value?      <anydata>
```

1) Reconfiguration

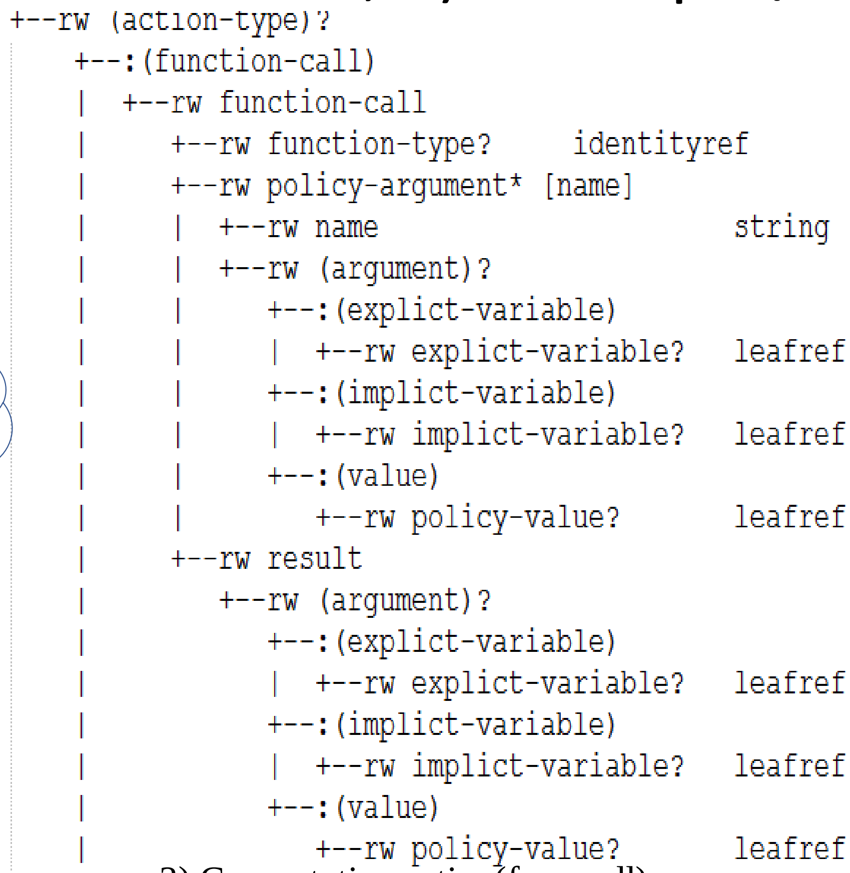
Do we need to integrate CRUD capability and YANG Path capability for moving, insert?



```
+--rw (action-type)?
   +--:(logging)
   |  +--rw logging
   |     +--rw type?               logging-type
   |     +--rw policy-variable*?   leafref
```

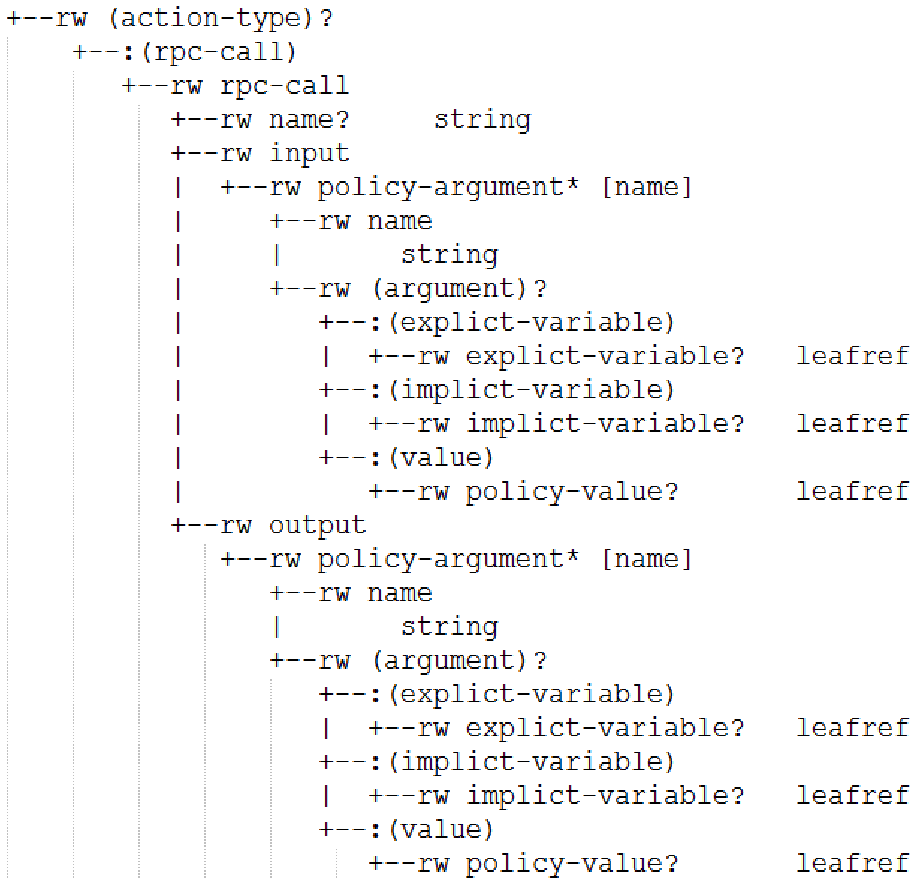
2) Notification

Open question: relation between script and ECA model



```
+--rw (action-type)?
   +--:(function-call)
   |  +--rw function-call
   |     +--rw function-type?     identityref
   |     +--rw policy-argument* [name]
   |     |  +--rw name                         string
   |     |  +--rw (argument)?
   |     |     +--:(explict-variable)
   |     |     |  +--rw explict-variable?   leafref
   |     |     +--:(implict-variable)
   |     |     |  +--rw implict-variable?   leafref
   |     |     +--:(value)
   |     |        +--rw policy-value?       leafref
   |     +--rw result
   |        +--rw (argument)?
   |           +--:(explict-variable)
   |           |  +--rw explict-variable?   leafref
   |           +--:(implict-variable)
   |           |  +--rw implict-variable?   leafref
   |           +--:(value)
   |              +--rw policy-value?       leafref
```

3) Computation action(func call)
e.g., A+B-C or A+B*C, or A^2
Open question: where to store computation results?

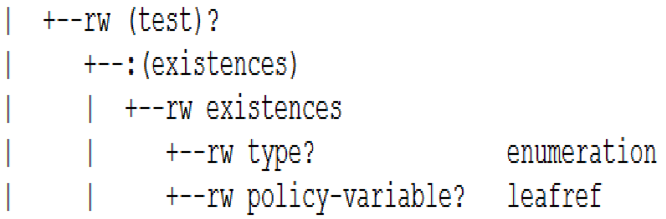


```
+--rw (action-type)?
   +--:(rpc-call)
      +--rw rpc-call
         +--rw name?      string
         +--rw input
         |  +--rw policy-argument* [name]
         |     +--rw name
         |     |      string
         |     +--rw (argument)?
         |        +--:(explict-variable)
         |        |  +--rw explict-variable?   leafref
         |        +--:(implict-variable)
         |        |  +--rw implict-variable?   leafref
         |        +--:(value)
         |           +--rw policy-value?       leafref
         +--rw output
            +--rw policy-argument* [name]
               +--rw name
               |      string
               +--rw (argument)?
                  +--:(explict-variable)
                  |  +--rw explict-variable?   leafref
                  +--:(implict-variable)
                  |  +--rw implict-variable?   leafref
                  +--:(value)
                     +--rw policy-value?       leafref
```

4) RPC input output (same as one invoked by client)
e.g., Add or remove subscription using RPC

Proposal: How to use PVs in the ECA Condition

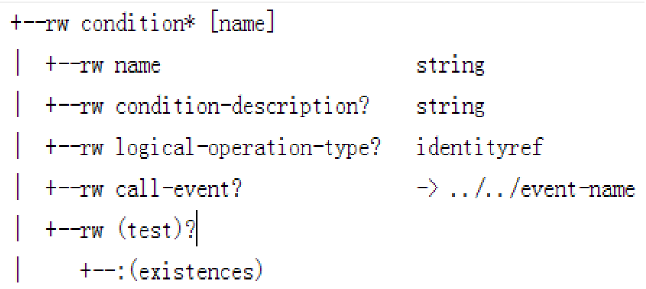
- How the client can use PVs in 1) condition evaluation



```
|   +--rw (test)?
|      +--:(existences)
|      |  +--rw existences
|      |     +--rw type?              enumeration
|      |     +--rw policy-variable?   leafref
```

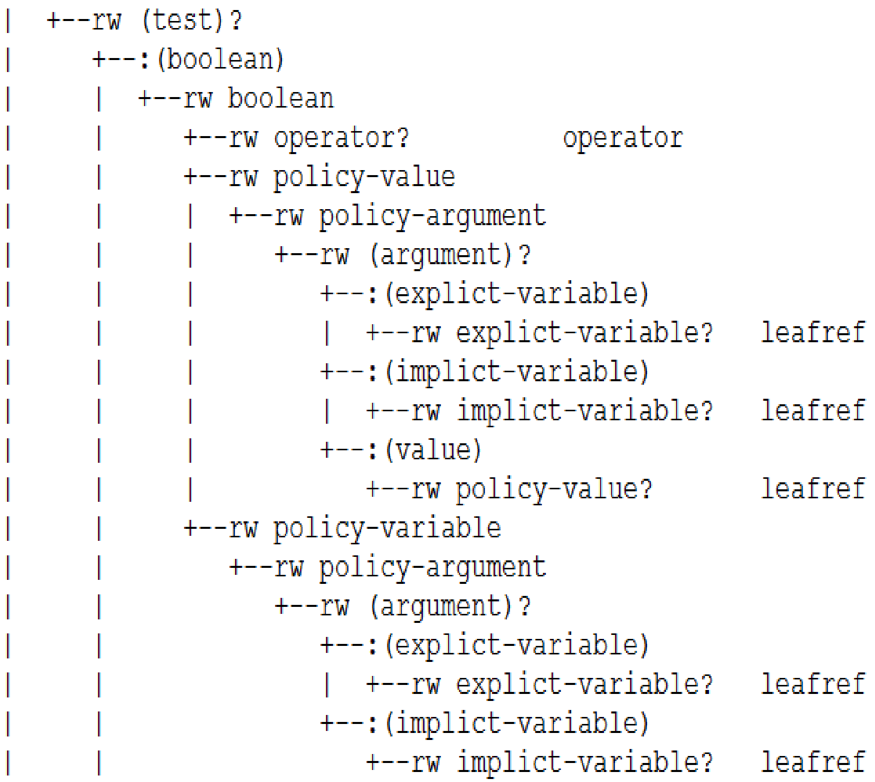
1) Existing Condition

Condition 1: arg exist (dispear, appear, change)



```
+--rw condition* [name]
|  +--rw name                      string
|  +--rw condition-description?    string
|  +--rw logical-operation-type?   identityref
|  +--rw call-event?               -> ../../event-name
|  +--rw (test)?
|     +--:(existences)
```

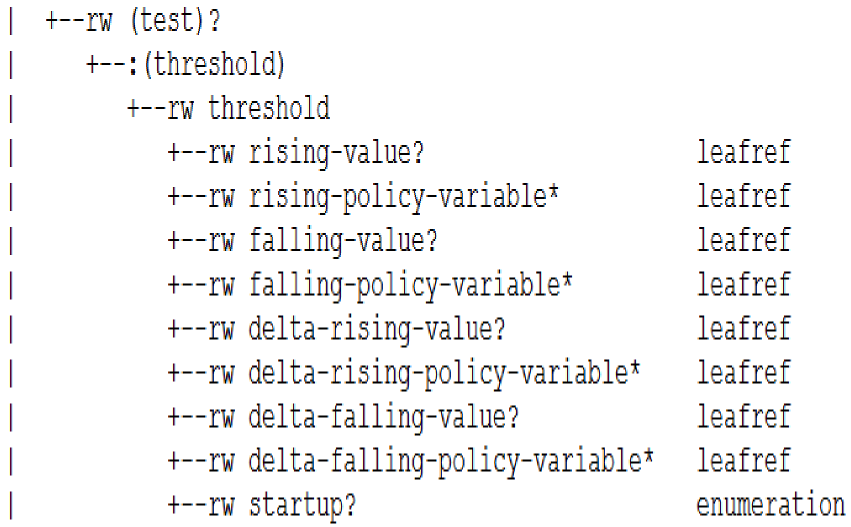
Condition 4: condition A and Condition B or Condition C



```
|   +--rw (test)?
|      +--:(boolean)
|      |  +--rw boolean
|      |     +--rw operator?          operator
|      |     +--rw policy-value
|      |     |  +--rw policy-argument
|      |     |     +--rw (argument)?
|      |     |        +--:(explict-variable)
|      |     |        |  +--rw explict-variable?   leafref
|      |     |        +--:(implict-variable)
|      |     |        |  +--rw implict-variable?   leafref
|      |     |        +--:(value)
|      |     |           +--rw policy-value?       leafref
|      |     +--rw policy-variable
|      |        +--rw policy-argument
|      |           +--rw (argument)?
|      |              +--:(explict-variable)
|      |              |  +--rw explict-variable?   leafref
|      |              +--:(implict-variable)
|      |                 +--rw implict-variable?   leafref
```

1) Boolean Condition

Condition 2: arg 1 (Policy variable) relation arg 2 (Policy value)

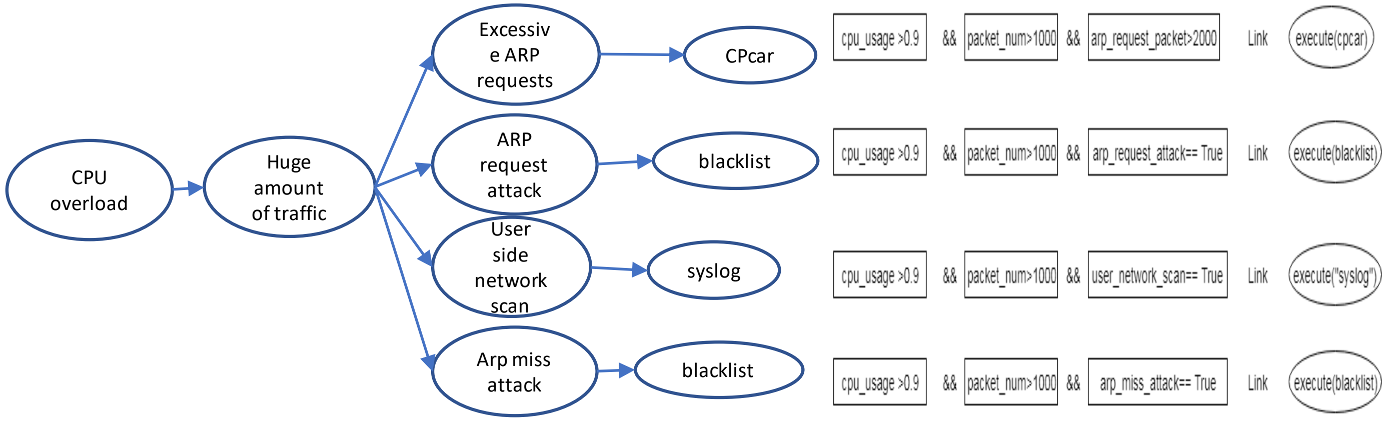


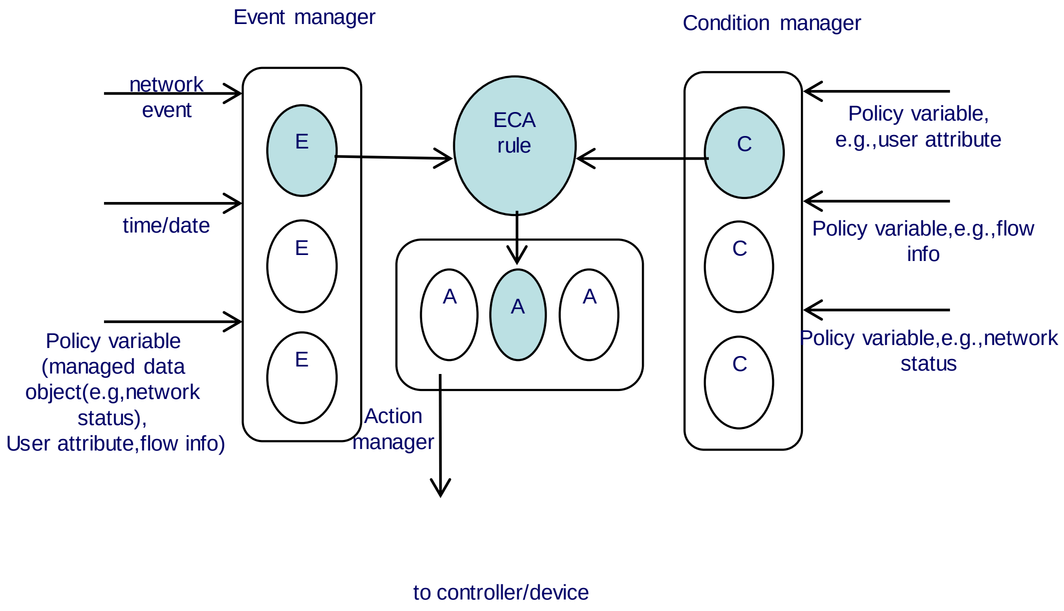
```
|   +--rw (test)?
|      +--:(threshold)
|         +--rw threshold
|            +--rw rising-value?                   leafref
|            +--rw rising-policy-variable*         leafref
|            +--rw falling-value?                  leafref
|            +--rw falling-policy-variable*        leafref
|            +--rw delta-rising-value?             leafref
|            +--rw delta-rising-policy-variable*   leafref
|            +--rw delta-falling-value?            leafref
|            +--rw delta-falling-policy-variable*  leafref
|            +--rw startup?                        enumeration
```

1) Threshold Condition

Condition 3: d (arg 1)/d (time) relation arg 2 (Policy value)

ECA Model Usage Example A



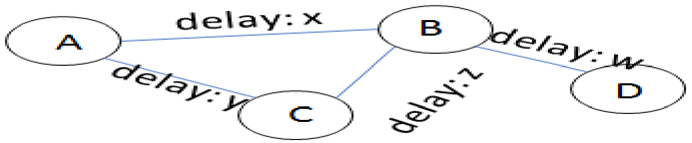


Event manager
Condition manager
network event
time/date
Policy variable (managed data object(e.g,network status), User attribute,flow info)
E
E
E
ECA rule
A
A
A
Action manager
to controller/device
C
C
C
Policy variable, e.g.,user attribute
Policy variable,e.g.,flow info
Policy variable,e.g.,network status

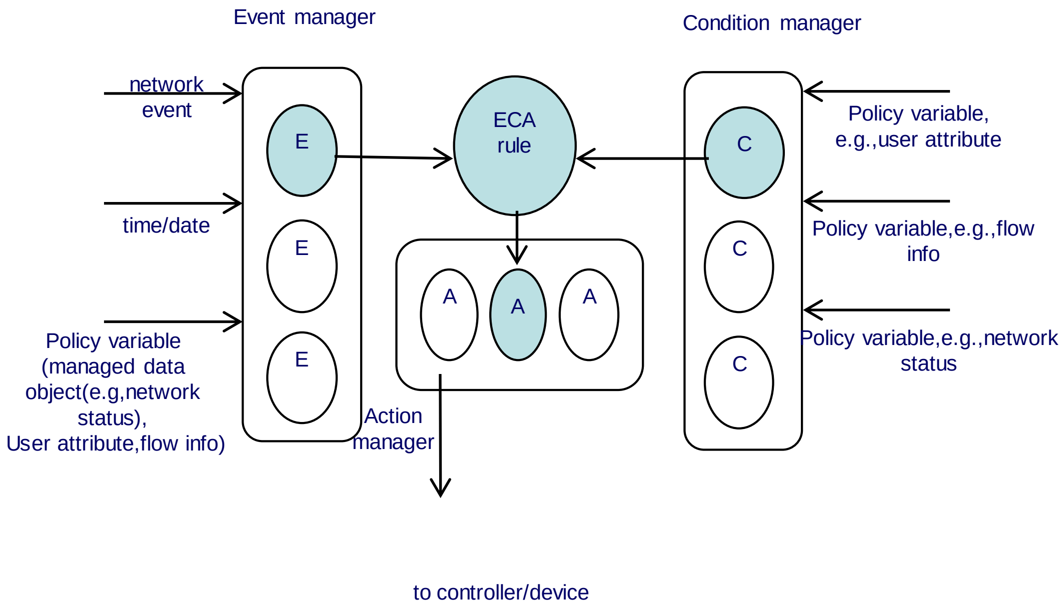
- Event: CPU overload
- Policy variable:
  - Variable 1: cpu_usage
  - Variable 2: packet_num
  - Variable 3: arp_request_packet
  - Variable 4:arp_request_attack
  - Variable 5: user_network_scan
  - Variable 6: arp_miss_attack
- Condition:
  - Condition 1: Cpu_usage>0.9&&packet_num>1000&&arp_request_packet>2000
  - Condition 2: Cpu_usage>0.9&&packet_num>1000&&arp_request_attack==true
  - Condition 3:Cpu_usage>0.9&&packet_num>1000&&user_network_scan==true
  - Condition4: Cpu_usage>0.9&&packet_num>1000&&arp_miss_attack==true
- Action:
  - Action 1: configure control plane committed access(CPCAR)
  - Action2: write into blacklist
  - Action 3: Syslog
  - Action 4: write into blacklist

ECA Model Usage Example B

3. TE path computation
Example



Example of policy: if(service_destination matches 10.132.12.0/24) Use path:
A=> B => D.
else Compute path with minimal delay.


Event manager
Condition manager
network event
time/date
Policy variable (managed data object(e.g,network status),
User attribute,flow info)
E
E
E
ECA rule
A
A
A
C
C
C
Policy variable,
e.g.,user attribute
Policy variable,e.g.,flow info
Policy variable,e.g.,network status
Action manager
to controller/device

- Event: TE Path computation
- Policy variable:
  - Variable1: service_destination
  - Variable2: src
  - Variable 3:dst
  - Vriable 4:e2e-path
- Condition:
  - Condition 1: service_destination matches 10.132.12.0/24
  - Condition 2: service_destination mismatches 10.132.12.0/24
- Action:
  - Action 1:Set path A=>B=>D
  - Action 2: call RPC for path computation with minimal delay
    - Input: src =A, dst=d
    - Output: e2e-path = a=>c=>d