Contents

[1 Introduction 3](#_Toc29937908)

[2 Technologies to consider 3](#_Toc29937909)

[2.1 Ontology for describing IoMT service conditions (Proposed by Konkuk University) 3](#_Toc29937910)

[2.1.1 General 3](#_Toc29937911)

[2.1.2 Examples 3](#_Toc29937912)

[2.1.3 Expression via RDF 4](#_Toc29937913)

[2.2 Ontology for describing IoMT setup information (Proposed by Konkuk University) 5](#_Toc29937914)

[2.3 Haptic Ontology for IoMT 5](#_Toc29937915)

[2.3.1 General 5](#_Toc29937916)

[2.3.2 Two-point threshold 6](#_Toc29937917)

[2.3.3 Consecutive Vibration 7](#_Toc29937918)

[2.3.4 Proposal on Vibro-haptic Ontology 7](#_Toc29937919)

[2.3.4.1 Possible expression of vibro-haptic technology 7](#_Toc29937920)

[2.3.4.2 Composition of Vibro-haptic Ontology 7](#_Toc29937921)

# Introduction

This document contains technologies proposed for MPEG-IoMT (ISO/IEC 23093) that provide the usage of ontologies and finite state machines for the management of MThings. When the sufficient description is provided by each proponent, technologies will be included in the MPEG-IoMT standard.

# Technologies to consider

# Ontology for describing IoMT service conditions (Proposed by Konkuk University)

# General

In MPEG-IoMT, the service description of MThing means information that MThing can serve. For example, users can check the service description of an MThing, such as price, usage condition, and connect to an MThing with a reasonable condition. Even though APIs, which inquire cost for use, exist in MPEG-IoMT Part 3, the return value of API is a simple float value that cannot explain complicated conditions like the service description. A tool for describing complicated conditions is necessary but there is no tool for describing the service description of MThing in MPEG-IoMT yet. So, we made an example related to the service description for proposing service description tools in this contribution.

# Examples

We created an example for the expression of service description of an IoMT Camera. Below is an example of MCamera’s service description including information such as charge, the permission of output according to functionality:

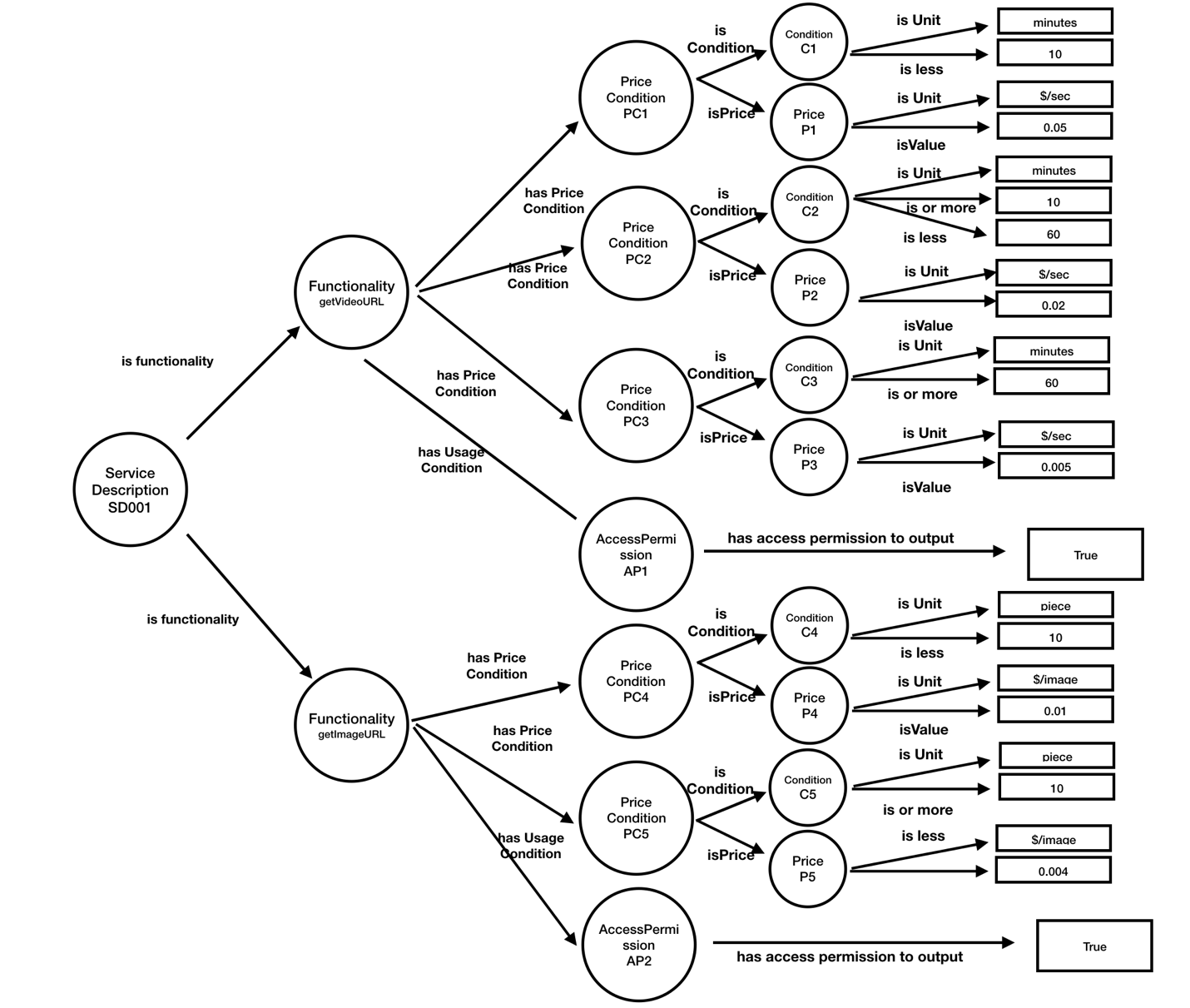
* MCamera
  + Functionalities
    - getVideoURL
      * Price
        + $0.05 per second less than 10 minutes (if x < 10 min)
        + $0.02 per second between 10 and 60 minutes (if 10 min <= x < 60 min)
        + $0.005 per second more than 60 minutes (if x > 60 min)
      * Access permission to output (The output is permitted to broadcast)
        + True
    - getImageURL
      * Price
        + $0.01 per image (if x < 10 pieces)
        + $0.004 per image (if x >= 10 pieces)
      * Access permission to output (The output is permitted to broadcast)
        + True

Below is an example of MDisplay. The MDisplay can be used for advertisement and the access permission to output is not necessary.

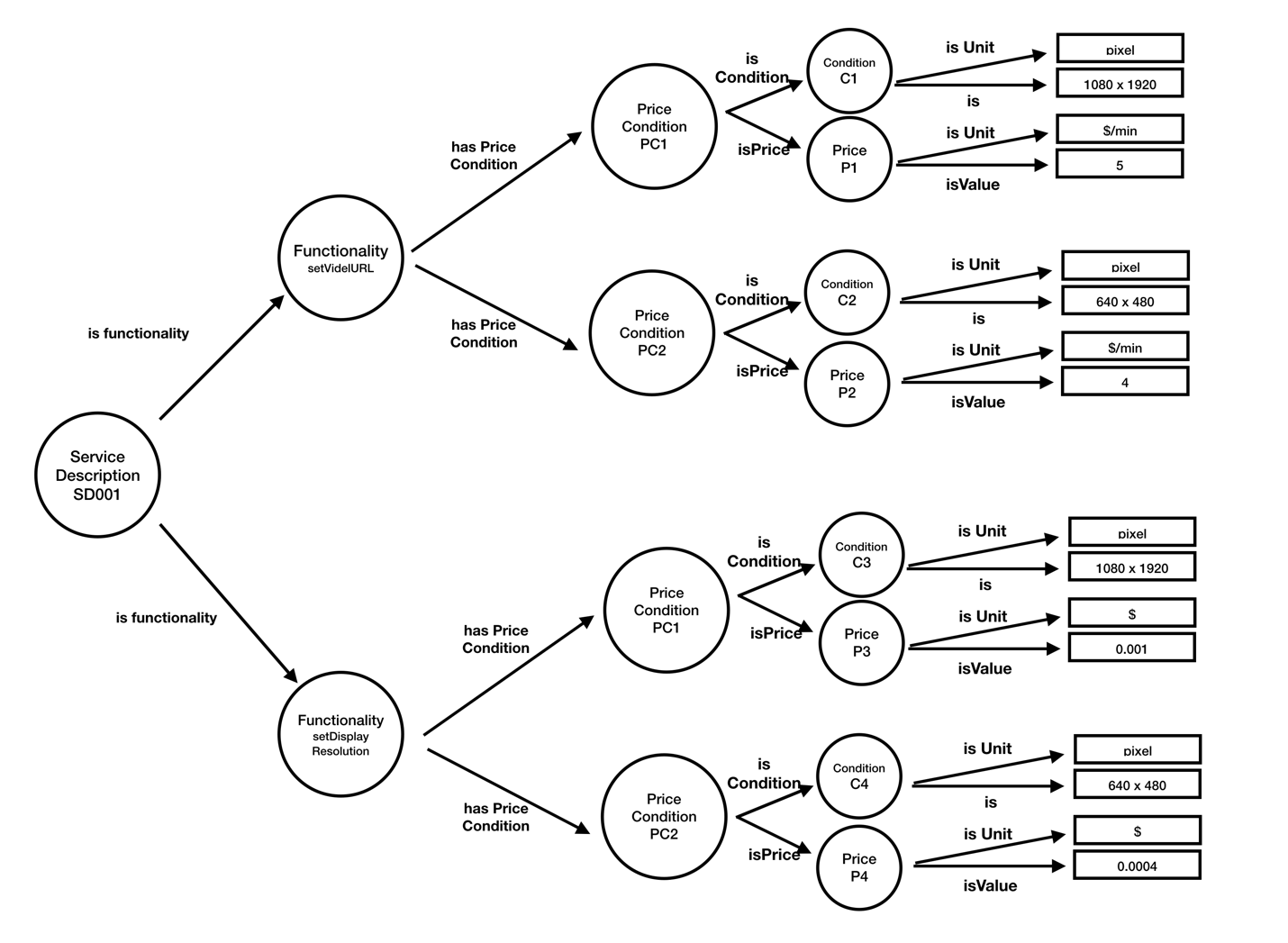
* MDisplay
  + Functionalities
    - setVideoURL
      * Price
        + $5 per minute if resolution is 1080 x 1920
        + $4 per minute if resolution is 640 x 480
    - Se\n
      * Price
        + $0.001 if resolution is 1080 x 1920
        + $0.0004 if resolution is 640 x 480

# Expression via RDF

The service description can be represented by the Resource Description Framework(RDF). The RDF is a tool for expressing resources by the circle node and atomic value by the square node with its property written on the arrow. Below figures are the RDF representation of service descriptions described in section 2.2.



**Figure 1. Service description expression of MCamera**



**Figure 2. Service description expression of MDisplay**

With these graphs, users can check information, which is a service description, of MThing and reckon which MThing has a better service description to the user. These graphs, RDF, can be uploaded into blockchain and users can look the information online with easy.

# Ontology for describing IoMT setup information (Proposed by Konkuk University)

TBD

# Haptic Ontology for IoMT (Proposed by ETRI)

# General

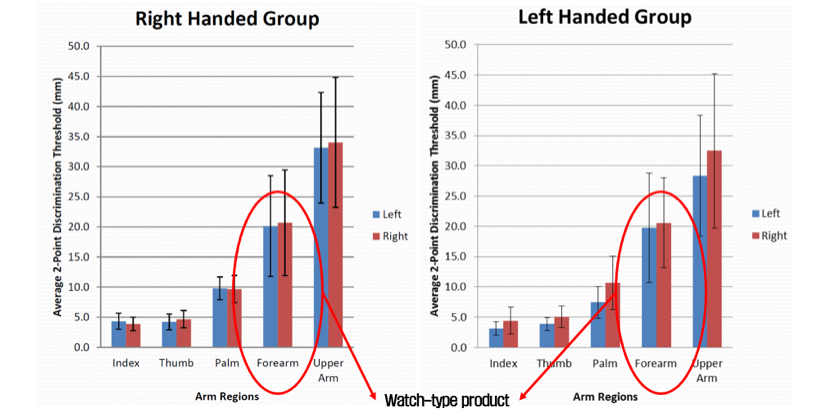
Korea is currently facing an aging society phenomenon, similar to the ones already encountered in many advanced countries. Elderly people are suffering from declining of audiovisual senses rather than tactile feeling. Hence, information delivery via the tactile sense would be a promising approach to enhance the day-by-day comfort of such people. Moreover, vibration is useful not only for elderly people but for seeing/hearing impaired people as well.

A smart vibration device conveys information by stimulating human skin. This type of vibration is useful for informing some situations like ringing a bell, receiving a text, noticing an appointment, etc. However, such vibration does not give what the information is, but just signal some important event occurred, thus being rather a simple alarm than an information transmission procedure.

With this contribution, we propose a novel technology to increase the information conveyed via vibration. We introduce a concept of a two-point threshold defined as a minimum distance allowing people to recognize two stimulations when two different points on human skin are stimulated at the same time. Suppose an m by n grid vibration panel with m x n vibration elements. Then we can convey a variety of vibration codes if the distance between elements is over the two-point threshold. Moreover, by considering successive combinations of m x n vibration signals, much more information can be sent.

# Two-point threshold

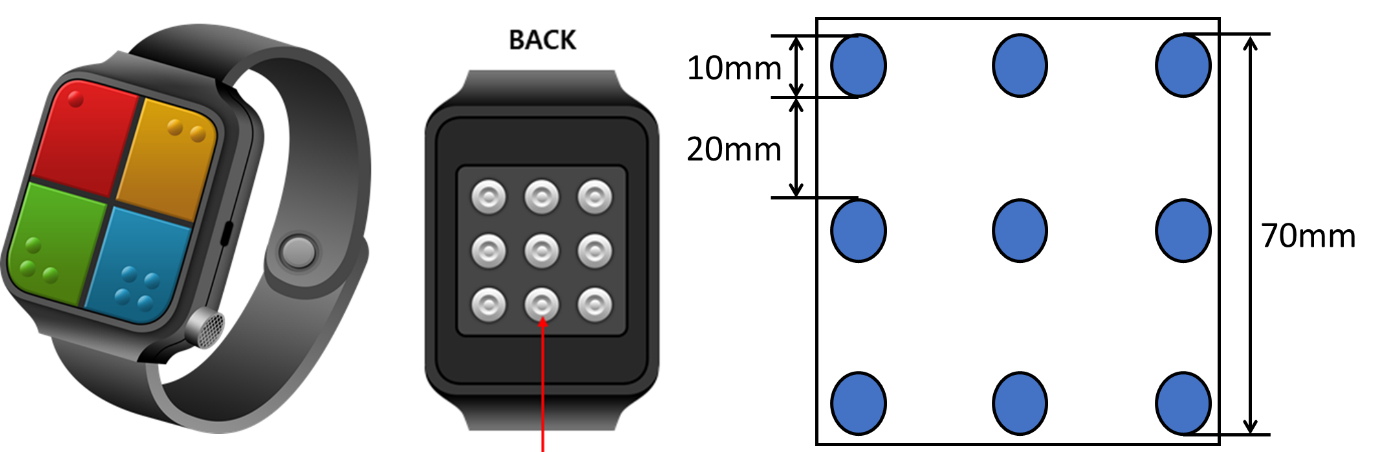
Our audiovisual capabilities are degraded as getting ages. In an aging society, the number of elderly people will increase day by day and we need to support their degraded capabilities. Vibration haptic technology is a challengeable candidate but its main current limitation is given by the very limited (a priori binary) quantity of information it is conveying.



**Fig. 1. Experimental results of two-point threshold**

Here we introduce a concept of a two-point threshold defined as a minimum distance to recognize two stimulations when two different points on human skin are stimulated at the same time. It is experimentally known that the two points thresholds for finger, forearm, thigh are 5, 20, 50mm respectively. Fig. 1 shows how the two-point threshold is measured for right-handed and left-handed groups [1]. It is approximately measured as 20mm in the case of the forearm.

When we produce a watch-type wearable haptic device with 3x3 grid vibration elements, the device need at least a 70mm x 70mm area which is hard to wear on the user’s forearm (See Fig. 2). So, if we want to manufacture that kind of wearable haptic device, we need to find solutions to reduce the two-point threshold as well as to minimize the size of the vibration element itself.



**Fig. 2. Instantiation of two-point threshold applied wearable haptic device**

# Consecutive Vibration

The principle is to reduce the two-point threshold by creating sequences of consecutive vibrations.

For example, a 3x3 grid vibration panel expresses the right direction by vibrating a left-bottom at t, a left-middle at t + 1, a left-top at t + 2, a middle-top at t + 3, and finally a right-top at t + 4. If we manufacture the grid panel as a wearable watch, the two-point threshold would be 10mm on the forearm. As shown in Fig. 3, we propose a consecutive vibration technology connected to a personal smart device so as to convey fluent haptic information with the reduced two-point threshold.



**Fig. 3. Instantiation of a consecutive vibration on wearable haptic device**

# Proposal on Vibro-haptic Ontology

# Possible expression of vibro-haptic technology

Possible vibration patterns can be defined according to specific purposes, as exemplified below:

1) ***Load guidance***: left-turn, right-turn, go-straight, slow-down, traffic signals

2) ***Danger warning***: crossing, manhole, obstacle, steep slope, under-construction

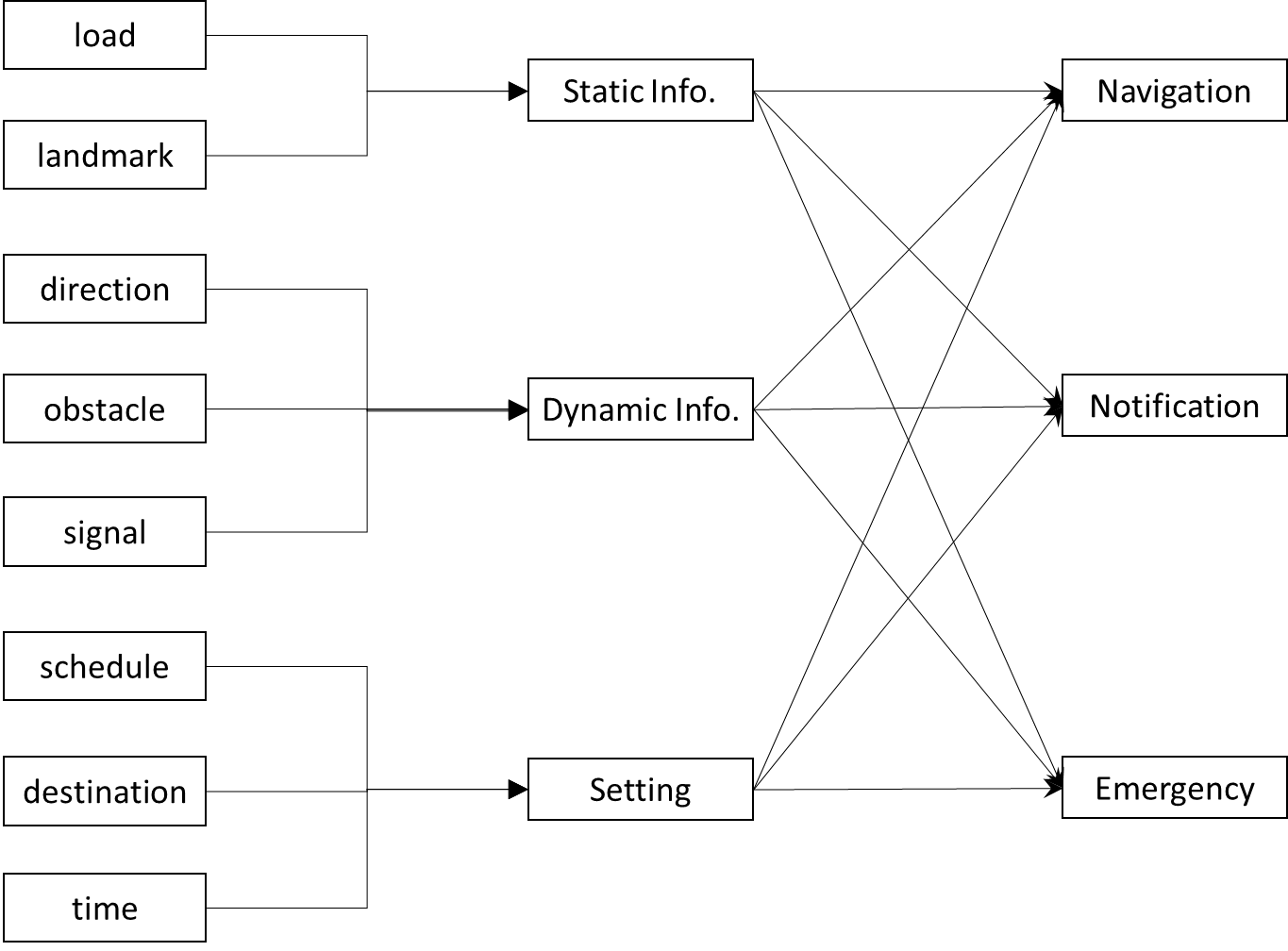
3) ***Event alert***: doorbell, traffic lights, bell sound, fire, earthquake

4) ***Alarm of a calendar event***: a drug-taking time, outing, regular notice of time, visit

5) ***Location***: pharmacy, convenience store, government office, my house, destination

# Composition of Vibro-haptic Ontology

With the possible expression of Vibro-haptic technologies, we can compose a Vibro-haptic ontology as shown in Fig. 5. The ontology can be extended more elements for the general purpose. Navigation, notification, the emergency could be a user’s purpose to accomplish. The individual purposes need some combination of static and dynamic information and user’s preferred settings. This information could be aggregated from several services like google map, traffic signal, weather casting.



**Fig. 4. Instantiation of Vibro-haptic Ontology**

# Mission diagrams for IoMT (Proposed by Konkuk Univ. and Myongji Univ.)

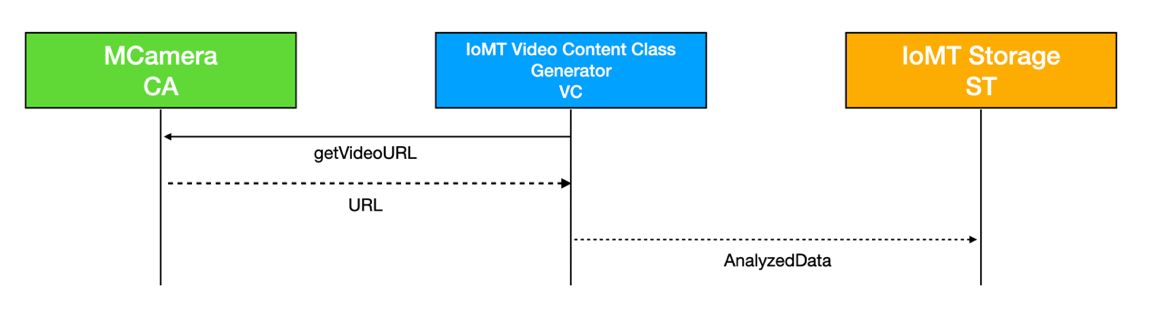
# General

In MPEG-IoMT, the mission means that what has to be done between MThings. Each MThing should know what state they are in, what they should do and which data they should transfer. However, there are no tools for describing the missions between MThings, and we start to discussion on the need of a new type of tools for representing missions of MThing.

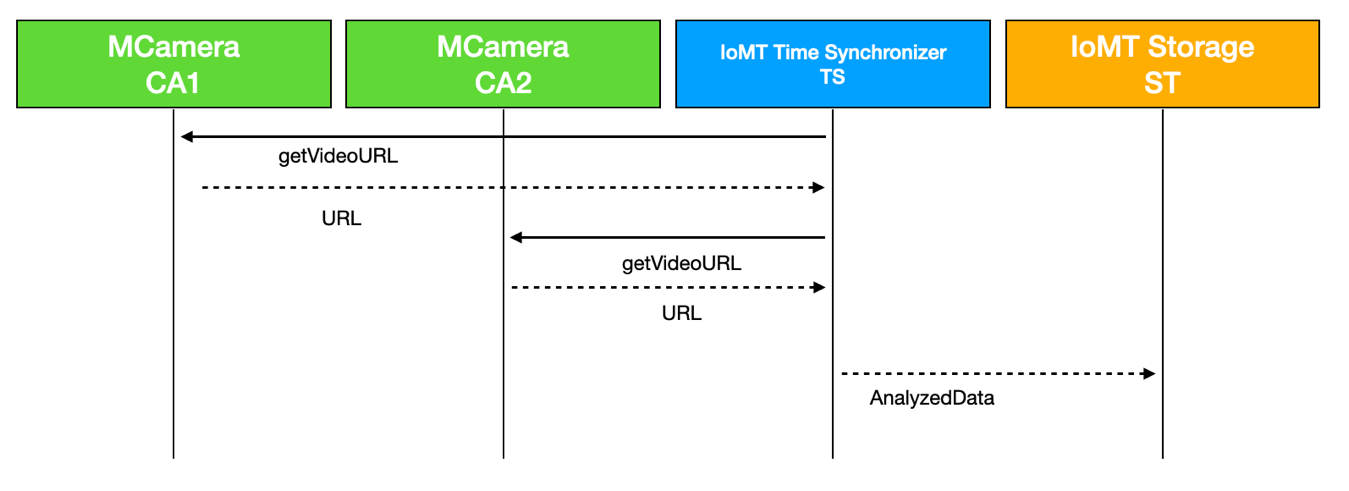
# Flows of MThing

 We created two examples related to MVideo Content Class Generator and MTime Synchronizer. First of all, it is necessary to understand of the flows of control and data between MThings. The figure below is representing about flows of APIs and data among MVideo Content Class Generator, MCamera, and MStorage.

**Figure 1. APIs and Data Flows of MCamera, MVideo Content Class Generator, and MStorage.**



For analyzing a video source, MVideo Content Class Generator has to obtain a video URL from MCamera. After analysis, analyzed data can be saved in MStorage through transmitting from IoMT Video Content Class Generator To MStorage. Figure 1 shows calling API and transmitting data between MThing but cannot represent API calls and data transfer inside an MThing and even current state of MThing.

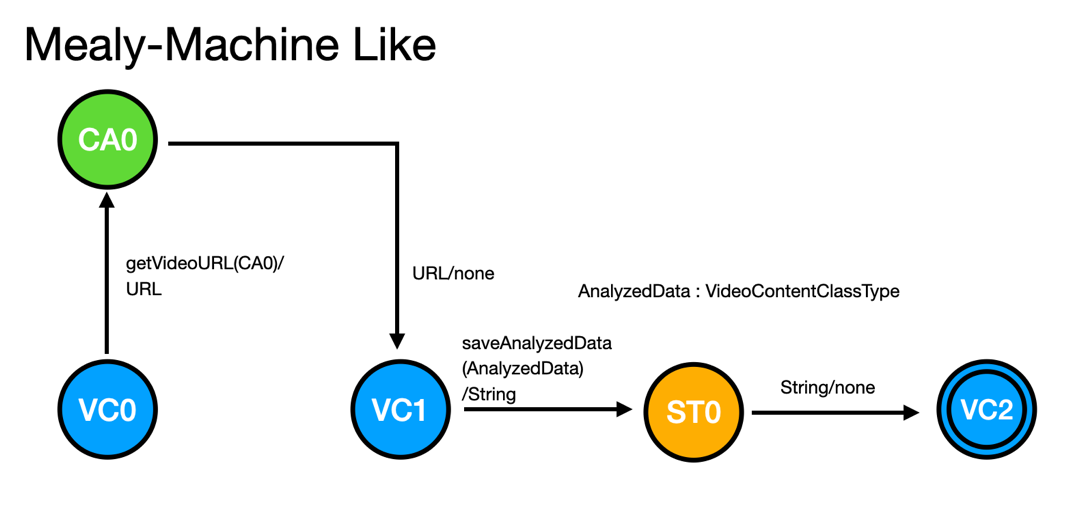


**Figure 2. APIs and Data Flows of MCamera, MTime Synchronizer, and MStorage.**

Figure 2 indicates communications among MTime Synchronizer, MCamera, and MStorage. For calculating a time difference between two videos, MTime Synchronizer has to acquire two video URLs from MCameras. MTime Synchronizer analyzes the time difference between the two videos and sends analyzed data to MStorage. Likewise, the call of APIs and transmitting data between MThings can be represented but it cannot be expressed that what state they are in, and internal API calls inside MThing.

# Mission Diagram of MThing

For illustration og a mission diagram of MThing, we borrow the Mealy Machine which is a finite-state machine whose output value is determined by input value and current state. By utilizing the Mealy Machine, the mission diagram can portray the current state of MThing, Internal APIs, inputs and output between MThing and even inside MThing.



**Figure 3. Mission Diagram of MCamera, MVideo Content Class Generator, and MStorage.**

 Figure 3 shows the mission diagram of figure 1 in the Mealy Machine form. The mission diagram represents each state by a circle node and shows input(left) and output(right) of each state next to the arrow, which shows the transition of the state. Input and output may be API in an MThing or data coming out of the MThing, or none. Color of the node represents the type of an MThing, in Figure 3, and 4, the blue represents MAnalyzer, the green represents MSensor and the orange represents MStorage.

In MPEG-IoMT, MCamera cannot send URL before another MThing requests a URL. So initial MVideo Content Class Generator has an initial state of mission diagram. If MVideo Content Class Generator requests a URL by calling getVideoURL APIs, MCamera returns a video URL. The output URL can be an input parameter for MVideo Content Class Generator. After analyzing the video source, MVideo Content Class Generator sends the analyzed data(generated content class of video) to MStorage. In order to save AnalyzedData in MStorage, calling saveAnalyzedData API by MVideo Content Class Generator is necessary. After saving the analyzed data, we can get a string which is a unique file number and finally the mission is finished.

**Figure 4. Mission Diagram of MCamera, MTime Synchronizer, and MStorage.**

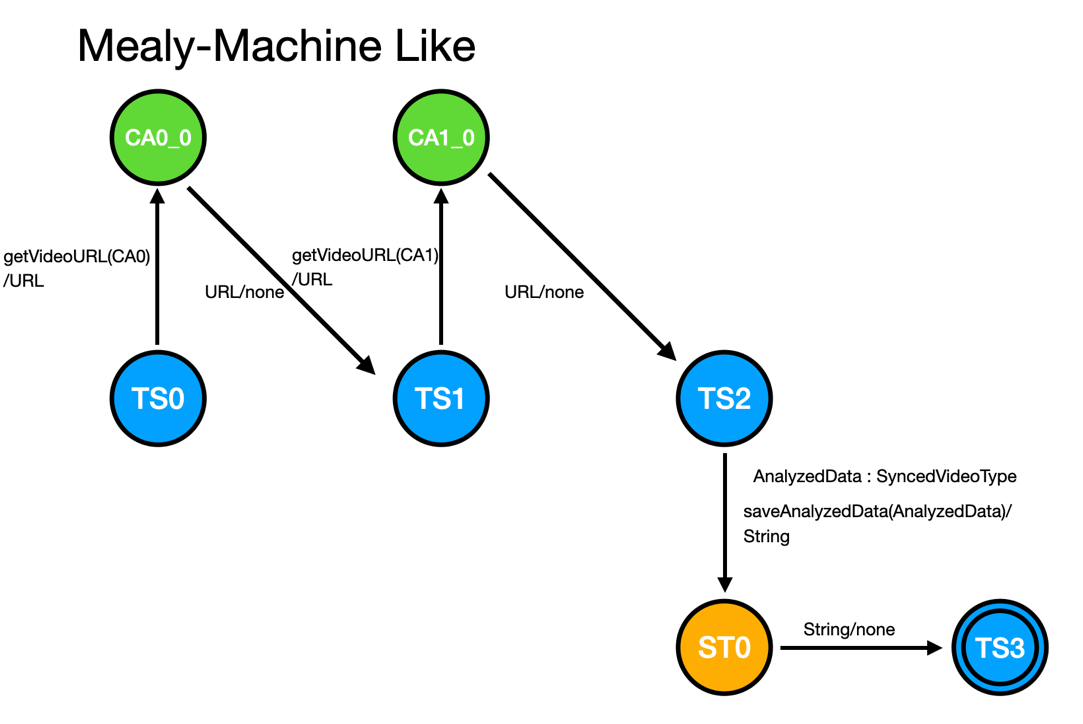


Figure 4 shows a mission diagram of MCamera, MTime Synchronizer, and MStorage. MTime Synchronizer requests MCamera to get a video URL like above MVideo Content Class Generator but there is a difference in here. MTime Synchronizer analyzes a time difference between two videos so needs two video URL. With only one URL, analyzing the video source cannot proceed, so it is necessary by writing two URLs over arrow (TS1 -> TS2) in Figure 4 to represent that MTime Synchronizer requires two video URL for analyzing time difference.  After MTime Synchronizer acquires two video URLs and analyzes the video sources, MTime Synchonizer analyzes the time difference of the video sources and sends AnalyzedData to MStorage for saving the data. MStorage saves the analyzed data through saveAnalyzedData API called by MTime Synchronizer and finally the process is terminated.