Device Description Language

1 DDL Data Nomenclature

1) Member and element names all start with upper case. Member and element names use a "_" to separate multiple word names so as to improve readability.
2) Member and element names avoid the use ofabbreviations to enhance readability.
3) Within this document, members are listed in a '[]' collection.
4) Within this document, elements are listed in a '{}' collection.

2 DDL Framework

2.1 "Devices"

"Devices" is a member of DDL file. It contains two kinds of elements. One is sensor and the other is actuator.

2.2 "Description"

"Description" is an element of device. It contains the identification information of a device.

2.2.1 "Name"

"Name" is an element of "Description". It defines the name of the device.

2.2.2 "Device_Type"

"Device_Type" is an element of "Description". It defines the type of the device.

2.2.3 "Verbose_Description"

"Verbose_Description" is an element of "Description". It shows the description of the device.

2.3 "Interface"

"Interface" is an element of device. It contains the description of the interface of the device. It contains two members: "Signal" and "Reading".

2.4 "Signal"

"Signal" is a member of "Interface". It contains the description of signals transmitted via the interface.

2.4.1 "id"

"id" is an elements of "Signal". It defines the signal id of the device. It starts with 's' and followed by a numerical number.

2.4.2 “Operation”

“Operation” is an element of “Signal”. It defines the operation of the signal. The example operation is input or output.

2.4.3 “Type”

“Type” is an element of “Signal”. It defines the type of signal. The example type is digital or analog.

2.4.4 “Measurement”

“Measurement” is an element of “Signal”. It defines the measurement of signal.

2.5 “Reading”

“Reading” is an element of “Interface”. It contains the information of readings converted from the signal.

2.5.1 “id”

“id” is an element of “Reading”. It defines the identification of reading.

2.5.2 “From”

“From” is an element of “Reading”. It defines the id of signal from which to read

2.5.3 “Type”

“Type” is an element of “Reading”. It defines the type of the device.

2.5.4 “Measurement”

“Measurement” is an element of “Reading”. It defines what the device measures. For a temperature sensor, it measures temperature. For a noise sensor, it measures noise.

2.5.5 “Unit”

“Unit’ is an element of “Reading”. It defines the unit of the reading result. For instance, the unit of temperature can be centigrade or fahrenheit.

2.5.6 “Data_Type”

“Data_Type” is an element of “Reading”. It defines the return data type of the lower level driver.

2.5.7 “Return_Length”

“Return_Length” is an element of “Reading”. It defines the length of the returning data of the lower level driver.

2.5.8 “Computation”

“Computation” is an element of “Reading”. It defines how to convert the data returned from the lower level driver. It contains two elements: “Type” and “Expression”. Usually type is formula. “Expression” is the detailed way to convert raw data.

3 DDL Parser

DDL is based on the concept that the availability of standard device descriptions allows for the development of intelligent environment applications integrating multiple sensors, actuators and complex devices. By representing each device as a service in the service-oriented architecture(SOA), programming the smart space becomes as easy as calling a number of methods from the device services.

DDL Parser helps parse the ddl file and generate device driver as much as possible. Our DDL Parser is implemented by Python which can import JSON file and make an analysis easily. The high level device driver include read function for sensor device and read write functions for actuator device. The high level device driver call the low level device driver and convert the data to a readable format for users.

4 DDL Structure(JSON):

```
Devices
    “Description”: {
        “Name”: // it defines the name of the device
        “Device_Type”: // it defines the type of the device
        “Verbose_Description”: // it shows the description of the device
    },
    “Interface”: {
        “Signal”: [
            {
                “id”:  // it defines the signal id of the device
                “Operation”: // it defines the operation of the signal, input or output
                “Type”:  // it defines the type of signal, digital or analog
                “Measurement”:  // it defines the measurement of signal
            }
        ],
        “Reading”: [
            {
                “id”:  // it defines the id of reading
                “From”:  // it defines the id of signal where to read
                “Type”:  // it defines the type of the device
```

```
                “Measurement”:  // it defines what the device measures
                “Unit”:  // it defines the unit of the reading result
                “Data_Type”:  // it defines the the return data type of lower level driver
                “Return_Length”:  // it defines the length of the return data
                “Computation”:
                        “Type”:  // it defines the type of computation
                        “Expression”:  // it defines the expression of computation
            }
        ]
}
```