Home Appliances & Entertainment (HAE) Service Framework

F2F Meeting in Beijing – June 23 ~ 26, 2015

Attendees

Name	***Company***
Inhwan Choi	LGE
Wonchul Choi	LGE
Seungchul Han	LGE
Stefano Toppan	Electrolux
Mauro Taiariol	Electrolux
Milton Wang	Haier
Lu Xiaoqiang	Haier
Yoshinori Nagai	Sharp
Tomoki Ogawa	Panasonic
Shigeki Nakamura	Sony
Takeshi Oishi	Sony
Jiahe Luo	Sony
Zhang Yi	TCL
Wei Zeyin	TCL

Day 1 (June 23rd) Meeting Notes

1. Discussion on shared interface – Display

   A. Members agreed to keep this Display interface as is and if the IRB review challenge necessity of this interface definition, then we can reconsider adding additional useful capability or information at that time.

   B. Responding to Mike's comments, Sony team and Inhwan Choi agreed that there is no need to reserve a special value to represent that the graphical resolution is unavailable because they believe all display has its own graphical resolution.

   C. Members agreed that this Display resolution information and its future possible settings such as brightness, contrast, and sharpness settings look similar to UserInterfaceSettings in some aspect, but they are not just about user interface settings. So members agreed to treat this

Display interface as one of other shared interfaces, and thus won't put this Display interface under the UserInterfaceSettings namespace.

2. Resolutions on General Open Points (ASAHAE-13)

    A. Open point 7

        i. Regarding to the frequency of property change signal emission issue, members preferred to keep this point as still open until Core WG's resolution will be received upon our request to support throttling mechanism or provide some clear guideline.

    B. Open point 10

        i. Inhwan Choi suggested adding human readable description property to each HAE device interface, which is configurable by users.

        ii. Takeshi Oishi commented that if a device has one device interface, that description property overlaps with the DeviceName of the About data, which is configurable by using Config. service.

        iii. Inhwan choi commented that this additional information could be used as additional descriptive information of sub-units different from just friendly name.

        iv. Members deferred making a decision on this point until end of discussion of general open points to have additional time to consider.

    C. Open point 11 (Updated after Day 2 meeting)

        i. Regarding to the implication of security annotation, two questions were raised.

            1. Is it possible or allowed to change the security annotation by device vendors when implementing that interface on devices ?

            2. If the security annotation of an interface is false, then is it possible or allowed to use non-AllJoyn security mechanism to make an access to that interface secure ?

        ii. Panasonic and Sharp would like to decide settings of security annotation based on the consultation results and take a lead to get consultation on this issue from IRB or Core WG.

        iii. At this moment, most of members agreed that it's better to set the security annotation as false unless there is no critical information that should be protected. But the consultation results is important for our final decision.

    D. Open point 12

        i. Regarding to the namespace structure issue, Inhwan Choi encourage members to think about the proposed structure (https://jira.allseenalliance.org/browse/ASAHAE-17) and provide any comments or better idea. A time slot is reserved for this discussion at Day 4.

    E. Open point 13

        i. All members agreed that defining standard alert messages as Mike suggested might be useful, but it will require a significant amount of time to reach consensus and thus, standard alert messages won't be developed for the initial release of service framework.

        ii. Inhwan Choi suggested to divide the alert code space into two blocks for future possibility to define standard messages. One is for standard alert messages, and the other is for vendor-specific or proprietary ones.

iii. In the future, once the standard alert messages are defined, then there might be some necessity to send the essentially same message by using different alert codes for backward compatibility. In that case, the alert messages can be sent two times with different alert codes. One is with standard alert code, and the other is with vendor-specific code for backward compatibility.

iv. All members agreed with this mechanism and the following division of alert code space is accepted.

   1. Standard alert messages : 0x0000 ~ 0x7FFF
   2. Proprietary or vendor-specific alert messages: 0x8000 ~ 0xFFFF

F. Open point 14

i. Inhwan Choi raised an issue that there is no cycle selection capability for washing and cooking device interfaces and it looks very ridiculous not to have any cycle selection mechanism from user experience point of view.

ii. Electrolux team expressed that it would be very difficult to standard common enumeration of cycle programs because different cycle names are used across different regions or countries for marketing purposes.

iii. Tomoki Ogawa agree on the necessity of having cycle selection mechanism but he commented that it is better to have very limited set of enumeration such as normal mode or else, because it is not easy to standardize commonly used cycle programs and it is hard to standardize common behavior.

iv. Inhwan Choi responded that even though the marketing name for a specific cycle program might be different across regions (for example, jeans for Europe, but cotton for North America), if they are essentially the same internal operation to each other, then it would be okay to provide standard cycle name and description, and it is fine to use different marketing names for the identical cycle program or id.

v. Some members commented that each vendor or each device might have unique cycle programs that can’t be covered by standard cycle program.

vi. Inhwan Choi suggested to divide the cycle id space into two pieces as we did for alert code messages. One is for standard cycles and the other is for vendor-specific or proprietary ones. The proprietary cycles are intended to be used only by device vendors, not by 3rd party control application developers.

   1. Standard cycles : 0x0000 ~ 0x7FFF
   2. Proprietary range: 0x8000 ~ 0xFFFF

vii. Members agreed with that mechanism and in this way, we will define OperationalCycle property for monitoring purpose. In addition, for enabling remote cycle selection, members agreed to add SupportedCycleSelections property.

viii. Tomoki Ogawa asked if a device can support only proprietary cycle identifiers or not.

ix. Mauro Taiariol and Inhwan Choi answered that it should be possible to support proprietary cycle identifiers.

x. Tomoki Ogawa asked if a device can support empty cycle identifier or not.

xi. Inhwan Choi answered that empty cycle identifier should be allowed because the device interface is a mandatory interface for that type of device.

xii. Inhwan Choi commented that we should consider this case: When a controller which doesn't know about proprietary cycle enumeration reads the proprietary cycle id as the current selected cycle, what should be reported to the user ?

xiii. Mauro Taiariol responded that that could be left as a decision to be made that control application. For example, vendor-specific cycle or unknown cycle program.

xiv. Regarding to the current cycle identifier, Mauro Taiariol commented that there might be no cycle selected at some specific point of time. For oven, initially there is no cycle selected.

xv. Mauro Taiariol suggested to reserve a special value for "no cycle selected" and members agreed to reserve 0x0000 for the "no cycle selected".

G. Open point 16

i. Regarding to General vs Specific naming convention, (OperationMode vs AirConditioningMode or OperationCycle vs WashingCycle), members prefer the general approach, if we could provide a clear definition of those terminologies.

ii. Inhwan Choi suggested we add definitions of those terminologies to the theory of operation section of the OperationalControl or service framework level theory of operation document.

iii. Members agreed to use 'CyclePhase' rather than 'CycleStage'. LG team will replace the 'CycleStage' of the RobotCleaner interface with 'CyclePhase'.

H. Open point 15

i. Inhwan Choi pointed out the problem of the multiple cavities oven case and proposed a slight modification to the device model structure. (refer to : 2nd comment on https://git.allseenalliance.org/gerrit/#/c/4780/1/docs/interfaces/org.alljoyn.Hae.Device/Oven-v1.md)

ii. Inhwan Choi suggested to use the common interface as a container interface for sub-units and also suggested to define the oven as a single cavity device. In addition, he also pointed out that the current Refrigerator interface is nothing but container interface, and it is not needed any more because the common interface will serve as a container.

iii. Members reached a consensus that we remove the Refrigerator device interface and keep the Fridge and the Freezer device interface.

iv. Electrolux team asked that an oven might have two represented temperatures. One is for oven cavity and the other is for food probe. Electrolux team asked that in this case how we can describe AllJoyn bus object and identify the semantic meaning of the food probe temperature.

v. Inhwan Choi commented that defining the FoodProbe device interface could be a solution for this issue.

I. Open point 17

i. Inhwan Choi raised an issue about 3 different approach to 'AutoMode'. The first is separate interface approach, the second is the one of enumeration mode approach and the last is the separate mode property approach. Members agreed to revisit this issue tomorrow morning after having some time to think about.

Day 2 (June 24th) Meeting Notes

1. Review the draft Day1 meeting minutes.
   - A. Members reviewed and had additional discussion on open points 11.
   - B. Day 1 meeting minutes are updated to reflect the additional discussion.
   - C. All members approved Day 1 meeting minutes.
2. Revisit open points
   - A. Open point 10
     - i. Members disagreed with the idea of standardizing the position enumeration, because it will fall short in the long run.
     - ii. Regarding to adding the human readable description property, Inhwan Choi explained that the Configuration service framework can be reused to configure the property by appending additional fields to the configuration map data, which corresponds to the added description properties of each device interface and he added that if a device vendor doesn't want to make end-users configure that description, it just doesn't provide the additional fields.
     - iii. Members realized that the data type of human readable description property needs to be an array of strings to support multiple languages. In addition, using Configuration service framework doesn't require any property to be defined and so, we don't have to define any explicit property for human readable descriptions.
     - iv. Since there was some members who do not see much value from human readable descriptions and members realized that configuration service framework doesn't require properties to be defined, Inhwan Choi withdrew his suggestions to add human readable description property and use the configuration service framework.
     - v. Members reached a consensus that there is no fundamental solution and doing nothing is our resolution.
   - B. Open point 17
     - i. Regarding to the AutoMode, Inhwan Choi explained that we can categorize into 2 cases in terms of object characteristics to be controlled. One is the enumeration of operation modes and the other is the level of resource that can be represented by numbers such as wind strength, wind direction, and humidity level, etc.
     - ii. For the first case, defining the AutoMode as one of enumeration is fine. The AirConditioner falls under this category.
     - iii. Regarding to the second case, Inhwan Choi introduced 3 possible solutions
       1. Separate interfaces like AutoWindStrength, AutoHorizontalWindDirection, AutoVerticalWindDirection, AutoHumidity including AutoMode into each AutoMode interface
       2. One generic interface with a property to read the resource type to control and a method to enable/disable AutoMode.
       3. AutoMode as a special level. 0 : AutoMode, 1 ~ MaxLevel : numeric level.

iv. The disadvantage of the first solution is that we will end up with a lot of interfaces just for automatic control. They are not reusable by other resource type.

v. The disadvantage of the second solution is that whenever we want to control new resource type automatically, the interface definition needs to be updated.

vi. The disadvantage of the third solution is that when the AutoMode is selected, there is no information available on the automatically changing level..

vii. Members wanted to embed the AutoMode in each interface that requires AutoMode because the semantic meaning is clear and they are coherent resource/capability to each other.

viii. Thus the final resolution for the second case (automatic level control) will be embedding the AutoMode capability as a separate property/method with “byte” data type having a special value (255) for capability information, “Unsupported”.

3. Review Refrigerator device interface

   A. The Refrigerator device interface will be removed according to our resolution on open point 15.

   B. Electrolux team commented that the "RemoteControl" property "org.alljoyn.Hae" interface gives the possibility to disable the control operations for the entire device from remote device. It means that when "RemoteControl" is disabled, any methods of HAE interfaces which has control purpose (e.g. set methods) shall raise an error when invoked.

   C. Inhwan Choi requested to add that issue and proposed solution as one of general open point in the JIRA in order to share other members who cannot attend this meeting.

   D. Electrolux team suggested that we define a new pure container device interface rather than using the common interface as a container because the container interface should be one of device interface (generic type) to be associated with “Shared” interfaces to represent optional capabilities of the whole device, and if there are two devices under the root bus object for just a simple device, it looks strange.

   E. Members agreed with Electrolux team’s suggestion. To avoid any confusion or unnecessary complexity, two explicit rules are added. 1) Use of the container device interface for just one sub-unit shall be prohibited. 2) Children of children device shall be prohibited.

4. Review fridge device interface

   A. Members asked what is the meaning of SuperMode and requested to provide more detailed description of SuperMode.

   B. Members commented that since the SuperMode is not commonly available by fridge devices from different vendors, the SuperMode needs to have a special value for “Unsupported”.

   C. There was a lengthy discussion on the name of SuperMode and members preferred RapidMode to SuperMode.

   D. By the same reasons with AutoMode, this RapidMode was determined to be embedded in each device interface, because RapidMode can be better explained in the context of each device interface and it’s not generally applicable to most devices.

   E. Some members opposed to define a minimum required shared interfaces for each device.

   F. Inhwan Choi commented that we already reached a consensus to define minimum requirements for each device type, and if we do not define a minimum required shared interfaces, no common capabilities can’t be expected for the same standard device type.

G. After a lengthy discussion, members agreed to develop the minimum mandatory requirements on shared interfaces case by case (device by device).

H. Panasonic, Sharp and Sony opposed to having TargetTemperature and RepresentedTemperature as minimum mandatory requirements for the Fridge because they believe the temperature information of fridge doesn't provide useful information.

I. All members agreed to have DoorStatus as a minimum required shared interface.

5. Review freezer device interface

   A. It's same with the Fridge.

6. Review ice maker device interface

   A. There was a comment that the "IceMakerMode" looks like state rather than mode and Electrolux team confirmed this.

   B. Tomoki Ogawa commented that the "IceMakerMode" looks like proprietary states rather than commonly used states across device vendors.

   C. After a lengthy discussion, Electrolux team withdrew "IceMakerMode" because the general operational state such as On and Off can be provided by using OperationalControl interface and this "IceMakeMode" is a more detailed sub state information.

   D. Tomoki Ogawa suggested to add water level (in tank) property which value ranges from 0 to MaxLevel and Electrolux team suggested to add another property for water supply source (tank or pipe).

Day 3 (June 25th) Meeting Notes

1. Review the draft Day2 meeting minutes.

   A. All members approved Day 2 meeting minutes.

   B. Stefano Toppan asked if it's possible to return an error or not when org.freedesktop.DBus.NoReply is true.

   C. Inhwan Choi answered the IRB Guideline says "Methods defined with the org.freedesktop.DBus.NoReply annotation must not return anything." and if the NoReply annotation is true, even error message return is not allowed. (Refer to https://wiki.allseenalliance.org/irb/interface_design_guidelines_draft_1.1)

2. Review TV device interface

   A. Regarding to the minimum mandatory requirements on shared interfaces, Inhwan Choi suggested to add Channel and VolumeControl and remove OperationalControl and Display interfaces.

   B. Sony team disagreed to add the Channel interface as a minimum required shared interface for the TV because in the future, there might be purely internet based TV having no RF tuner and the concept of channel could be totally different.

   C. In the case of the VolumeControl interface, Sony team commented that if we don't have a separate interface for STB, we shouldn't add the VolumeControl interface as a minimum required shared interface for TV because some STB doesn't have the VolumeControl.

D. Inhwan Choi agreed that defining a separate device interface for STB is better because STB and TV have a different form factors from each other.

E. Sony team and LGE team agreed to have OperationalControl and VolumenControl interfaces as a minimum required shared interfaces for TV, and have OperationalControl interface as a minimum required shared interface for STB.

F. Sony team will upload updated TV device interface and new STB device interface.

3. Review washing machine device interface

A. Members clarified the name of terminologies and agreed to use OperationalCycle and OperationalMode rather than OperationCycle and OperationMode.

B. Members had a discussion on standardization of operational cycles for washing machine and some members (Panasonic, Electrolux and Sharp) expressed their concerns that it would be very difficult to define a list of standard cycle programs.

C. Inhwan Choi asked the members who expressed concerns “why do you believe it looks very difficult or unrealistic ?” and they answered the following reasons.

i. The cycle name on the 3rd party control application might be different from the one on the control panel of real washing machine because the cycle name on the control panel of real device might be different from device to device or region by region even referring to the same cycle. This issue is a more complex if we take into account that the display language can change. The fact that different strings are displayed make a confusion to the final users.

ii. User scenario of settings of operation cycle from remote application is unusual and thus, it is not much required.

iii. Looking at the below list of example cycle programs for washing machine (provided by LG), multiple different manufacturer’s own cycle programs might fall under the same standard cycle. Without any clear rule or criteria to define standard cycle programs, it would be really difficult to specify a standard set of cycles. Each manufacturer has a different rule or criteria to define their own cycle programs. Thus it is very difficult to deliver common set of standard cycles.
0 --- Normal : For washing everyday items like jeans, towels, shirts, sheets and mixed loads.
1 --- Heavy Duty : For washing heavily soiled cotton fabrics
2 --- Bulky/Large : For washing large items like blankets, comforter, pillows, and other items that have difficult absorbing water.
3 --- Bright Whites : For washing and maintaining bright white fabrics.
4 --- Sanitary : For washing the dirtiest clothing items like work clothes, baby diapers or heavily soiled garments.
5 --- Allergen : For reducing allergens including common triggers such as dust mites and pet dander.
6 --- Tub Clean : For cleaning a buildup of mold and germs in the air inside the washer tub.
7 --- Towels : For washing towels.
8 --- Permanent Press : For minimizing wrinkles for dress shirts and pants, wrinkle-free clothing and poly/cotton blend clothing.
9 --- Handwash/Wool : For washing shrinkable woolen or hand-washable items requiring delicate care.
10 --- Delicates : For washing clothes that can get damages easily.
11 --- Speed Wash : For washing lightly soiled clothing and small loads, this cycle can have clothes clean and fresh faster than a normal cycle.

12 --- Small Load : For washing small load.
13 --- Drain+Spin : For draining the tub and spinning clothes.

D. Inhwan Choi asked members how they think about operational modes for Air Conditioner

E. Members answered that for air conditioner, operational modes such as Heat, Cool, Dry, Auto, Economy are quite obvious to define so they believed it would be possible to define standard set of operational modes.

F. Inhwan Choi asked members how they think about operational cycles for Oven.

G. Members answered that Grill, Bake, Roast, Broil, Stream kinds of cooking cycles might be standardized. But Pizza, Salmon, Chicken kinds of cycles can't be defined.

H. Inhwan Choi concluded that the situation is different from device type to device type and members agreed.

I. Inhwan Choi suggested to keep the previous decision, try to define the standard list of cycle programs case by case and use the example cycle enumeration provided by LG as our starting point for washing machine. He agreed that we might end up with no standard cycle for washcing machine, but it should be accepted.

J. Mauro Taiariol suggested to have internal review by each company until next week.

K. Members reached a consensus that a list of selectable cycles should be provided and it has cycle identifiers for both standard and proprietary cycles. The set of exposed cycles shall be determined by device vendor's own decision. If the currently selected cycle is not exposed as one of selectable cycles and it is a proprietary cycle, then the cycle name can't be known to remote control application. To resolve this problem, we define two properties. One is SupportedOperationalCycles and the other is SelectableOperationalCycles. The SupportedOperationalCycles is a list of operational cycles supported by the home appliance device and the SelectableOperationalCycles is a list of operational cycles selectable by remote control application.

L. For only proprietary cycles, a mechanism to provide cycle name and its description with languageTag as an input argument will be provided.

M. The discussion continued to CyclePhase and members requested Electrolux to provide more detailed description. The enumeration of CyclePhase is accepted as it is.

N. Members commented that if washer doesn't support specific cycle phase, that cycle phase might not be displayed on the machine and the control application.

O. Members had discussions on cycle options for adjustment of cycle program.

P. Regarding to WaterTemperatureLevel, Inhwan Choi suggested to add a method to adjust the water temperature level for washing phase and members agreed.

Q. Members raised an issue about enumeration for qualitative representation such as hot, eco_hot, and eco_cold, etc because they could be very different from company to company.

R. Electrolux team suggested to define the qualitative representation as a numerical level that ranges from 0 to MaxLevel like the AirQualityLevel interface and members agreed.

S. Finally the followings are suggested and members agreed.

i. MaxWaterTemperatureLevel : y (0 means unsupported)

ii. WaterTemperatureLevel : y - 0 (lowest) ~ MaxWaterTemperatureLevel (highest)

iii. SelectableWaterTemperatureLevel : array of index (Empty array means only support monitoring, each cycle program can have a different list)

iv. GetWaterTemperatureInfo(languageTag) -> return array of string

1. Ex) (“sanity”, “hot”, “echo_hot” …)
2. Ex2) (“30 °C”, “60 °C”, “90 °C” …)

v. If there is water temperature level information, it is recommended for remote control application to show the string information to the end-users.

vi. The water temperature level information should be all or nothing promise, that means {“sanity”, “”, “hot”} is not recommended.

T. SpinSpeedLevel will be defined in the same way with the WaterTemperatureLevel.

U. LGE team suggested to add SoilLevel and NumberOfRinse.

V. Members agreed to add the SoilLevel but disagreed the NumberOfRinse because adjustment of number of rinse is not so common.

W. Members commented that explanation on the procedure of cycle selection and its adjustment for cyclic-operation devices needs to be added in the theory of operation section of that device interface.

Day 4 (June 26th) Meeting Notes

1. Review the draft Day3 meeting minutes.

   A. All members reviewed and approved Day 3 meeting minutes.

2. Review washing machine device interface

   A. Tomoki Ogawa commented that the OperationalTime couldn’t be included in a minimum required shared interfaces for washer because some washer just have a simple timer.

   B. Members agreed to keep only one OpertationalControl as a minimum required shared interfaces for washer.

   C. Inhwan Choi suggested treating the washer dryer in the same way with the washer if the washer dryer can be represented by the washer device interface.

   D. Electrolux team disagreed with Inhwan because washer dryer can’t be represented by only washer device interface and added that washer and dryer have some overlapping information between their device interfaces.

   E. Members reached a consensus on defining a separate WasherDryer device interface.

3. Review air conditioner device interface

   A. Tomoki Ogawa preferred to remove the WindStrength from the minimum required shared interfaces for air conditioner.

   B. Members agreed to remove the WindStrength and to keep 3 minimum required shared interfaces: OperationalControl, TargetTemperature, RepresentedTemperature.

C. Regarding to the operational modes, members agreed to take the same approach with the operational cycles.

D. LGE will update the AirConditioner device interface reflecting members' consensus.

4. Review robot cleaner device interface

    A. Inhwan Choi introduced why LG defined the robot cleaner as a cyclic-operation device.

    B. Inhwan Choi suggested removing "Resume" command of the OperationalControl because there is no distinction between "Resume" and "Start" command in real products.

    C. Members disagreed with that idea because that's just graphical issue, distinction might be helpful in managing internal state transitions.

    D. Inhwan Choi withdrew his suggestion to remove "Resume" command.

    E. Members recommended to use "Stop" command rather than adding new "ReturnBackToStation" command and LG agreed.

    F. Members reached a consensus to use "Idle" as a state that the robot cleaner is being charged, but the battery level is not above threshold to start its cleaning operation. If the battery charging level goes above that threshold, then the operational state will change from Idle to ReadyToStart automatically, and this additional example description should be added to the theory of operation section of the OperationalControl interface.

    G. Members recommended to keep 3 cycle phases such as "Cleaning", "Homing" and "Charging" to provide more detailed information of Working state.

    H. All members agreed to have the OperationalControl as a minimum required shared interface for the robot cleaner.

5. Review oven device interface

    A. The procedure to use oven was discussed and members found that there are two cases. In one case, user executes preheating by setting target temperature before cycle selection, but in other case, there is no preheating and cycle selection is the first step for cooking.

    B. Regarding to the cycle phase of oven, Electrolux team suggested using the same approach with the OperationalCycle because cycle phase of oven is very different from company to company or device to device. The following is the suggested solution.

        i. CyclePhase : byte

            1. Standard: 0x00 ~ 0x7F

            2. Proprietary: 0x80 ~ 0xFF

        ii. SupportedCyclePhases : array of bytes

        iii. GetCyclePhaseInfo(languageTag) -> return array of struct {index, cycle name}
        This will return information about only proprietary cycle phases.

    C. Inhwan Choi suggested adding preheating cycle option to the oven device interface because preheating cannot be one of cycle phase for oven.

    D. After Electrolux team updates the oven device interface, LGE team will post the proposed solution for preheating capability to the oven device interface.

E. Members agreed to remove the OperationalTime, TargetTemperature and RepresentedTemperature from the minimum required shared interfaces and thus, only the OpertionalControl is accepted as the minimum mandatory requirement.

6. Review common interface

A. Mauro suggested adding pure container device interface to the list of devices and LG will update the list of devices in the service framework level theory of operation document.

B. Members agreed to remove “DeviceType”, “DeviceModelName”, “DeviceManufecturer”, and “DeviceId” because they are redundant information that can be covered by About data.

C. Members raised an issue about necessity of error messages for the SetLocation method and LGE team agreed to remove error definition of the SetLocation method

7. Discuss on interface namespaces

(Refer to: https://jira.allseenalliance.org/browse/ASAHAE-17)

A. Inhwan Choi suggested that we revisit this issue after having some time to consider and encouraged members to share their idea via the Jira ticket.

8. Discuss on the constraint on the children of children device

A. Per Mike’s email comment, members had a discussion and they believed that it’s better to keep this constraint for the initial release of our service framework to ensure better interoperability and reduce implementation and testing cost.

9. Review electric fan and thermostat device interface

A. Members agreed to change meaning of 0(zero) of WindStrength value to “Off” and to use 1 as the lowest setting of continuous fan operation in order to represent the current fan operation of the thermostat when its fan auto mode is selected.

B. LGE team will update the WindStrength interface reflecting our decisions.

10. Review air purifier device interface

A. PowerfulMode (or TurboMode) will be included, but other optional capabilities are to be removed because they are not so common.

B. After additional research, Yoshinori will make a proposal on AutoMode inclusion if required.
(as one of enumeration of operational modes)

C. Yoshinori suggested to define additional shared interface “WaterLevel” to be used by Humidifier or possibly IceMaker.

D. OperationalControl is the only minimum required shared interface for air purifier.

11. Discuss on next steps.

A. To finish our work on interface definitions, Gerrit review shall be used as much as possible, and there will be no additional call per week.

B. Uploading new patchsets to reflect decisions made at F2F meeting : before 7/2 Call (T3)

C. Reviewing and casting Code Review votes : before 7/9 Call (T1)

D. Uploading new patchsets to resolve negative comments as quick as possible is always better.

E. Updated target submission date to IRB : just after 7/23 Call (T3)

F. We will keep the current weekly call sequence and thus, the next week meeting will be held on 7/2 T3 (Thu 5 am UTC).

G. To be efficient in our weekly meetings, the agenda items should be requested and discussed via mailing list before the meetings, and our meeting time should be devoted to only technical decisions or resolutions. Open discussions for technical resolutions should be done online as much as possible before the call.

H. How to submit our huge number of interface definitions should be asked to IRB members and this is Inhwan Choi's action item.

I. Getting answers on the security annotation related questions from AllJoyn security experts or IRB members should be done as early as possible and this is Tomoki Ogawa's action item.