CNT5517/CIS4930 Mobile Computing Atlas IoT Application IDE

HSIANG-YUAN LIAO, University of Florida

IDE Console Version Introduction

We provide a console-version of our IDE. It is developed in .NET CORE 3.1 in C#, and using the package Newtonsoft.Json to deal with Tweets. By using the commands for this IDE, users could receive IoT Tweets from Atlas Middleware through multicast group. After receiving enough Tweets, users could build IoT Applications according to the Services and Thing information which is captured by IDE in the Smart Space. The IDE also provides the feature to set relationships in an APP. And most importantly, user could activate the APPs by IDE. Some limitations and the features that could be added in the future are listed in the last paragraph.

Although the console version of the IDE is not that user friendly, it has the advantage to use on Linux/Unix platforms. And it has the potential for automation. Users could write a script to run it automatically because it is possible for a script to extract informations and enter commands/inputs through the console. It would also be easy to run multiple IDEs joining different multicast groups and get the access to different Smart Spaces concurrently. Also, it has the ability to be embedded into other software programs.

First, we will walk through some important commands by posting its command-execution result screenshots. Then we will explain how it works especially for **recipe** and **app** command. Below is the image after executing our command-line version IDE.

As the image shows, there are twelve commands for user. We will show them one by one below:

```
hsiang-yuanliao@AlexLiao-MBP IoTIDE % dotnet run


CNT5517 - Group 4, IDE Project Console Version



Please enter the commands below:
connect         (Start receiving Tweets through multicast group 232.1.1.1:1235)
disconnect      (Stop receiving Tweets and leave multicast group)
pause           (Pause to receive Tweets)
resume          (Resume to receive Tweets)
thing           (Show "Thing" information briefly)
entity          (Show "Entity" information briefly)
service         (Show "Service" information briefly)
relation        (Show "Relationship" information briefly)
showall         (Show all received Tweets in a brief format, for debugging)
recipe          (Build an APP by selecting 2 services and add a relationship for them)
app             (Activate APPs that has been finalized by recipe cmd)
exit            (Terminate this IDE program)
```

✦ connect

I. Create a UDP socket and join the multicast group 232.1.1.1 on port 1235

II. Display received Tweet from the multicast group

iii. Stores the received Tweets in program memory

iv. It won't stop until user disconnect it, user could manually check if there are enough received Tweets, it is more flexible

```
connect

Execute cmd connect

Successfully listening to multicast address group 232.1.1.1:1235
```

✦ disconnect

I. Leave the multicast group 232.1.1.1 on port 1235

II. If the received Tweets are enough, user could disconnect it first for performance concern

✦ pause

I. Pause to receive any Tweet if already connected

II. Pause to show received Tweet in consol

✦ resume

I. Resume to receive Tweet if already connected

✦ thing

I. Show a brief information for received Identity_Thing and Identity_Language Tweets

II. Both type of Tweets need to be received before showing its information

III. If the Identity_Language Tweet is not received for a particular Thing, it is pointless to send any service call to it. We need the Thing's IP address for service call.

```
[thing

Execute cmd thing:

[SpaceID]           [ThingID]           [IpAddr]            [Port]
Group4VSS           MyRPI_5341          192.168.3.55        6668

(Notice: Only the "Things" recevied their Identity_Tweet and Language_Tweet are shown)



Please enter the commands below:
connect         (Start receiving Tweets through multicast group 232.1.1.1:1235)
```

✦ entity

I. Show a brief information for received Identity_Entity Tweets

```
[entity

Execute cmd entity:

[SpaceID]           [ThingID]           [EntityID]          [Name]              [Description]
Group4VSS           MyRPI_5341          entityID1           Red Led             It is a red led
Group4VSS           MyRPI_5341          EntityID2           SoundSensor         A sound sensor detecting for l
Group4VSS           MyRPI_5341          EntityID3           LightSensor         A sensor for detecting if the
```

✦ service

I. Show a brief information for received Service Tweets

II. If there is not input or output in service API, it shows “void”

```
service

Execute cmd service:

[SpaceID]          [ThingID]          [Name]             [Input 1]               [Input 2]          [Output]
Group4VSS          MyRPI_5341         TurnOnRedLed       void                    void               void
Group4VSS          MyRPI_5341         TurnOffRedLed      void                    void               void
Group4VSS          MyRPI_5341         GetRedLEDStatus    void                    void               LED_Status
Group4VSS          MyRPI_5341         SoundDetection     PeriodOfTimeInSeconds   void               soundDetected
Group4VSS          MyRPI_5341         LightDetection     PeriodOfTimeInSeconds   void               isBright
Group4VSS          MyRPI_5341         BlinkRedLed        times                   void               void
```

✦ relation

I. Show a brief information for received Relation Tweets

```
relation

Execute cmd relation:

[Name]                       [Type]     [Service 1]        [Service 2]        [Description]
BlinkingProtectorON          interfere  TurnOnRedLed       BlinkRedLed        Don't turn ON  the RED Led while it is bli
BlinkingProtectorOFF         interfere  TurnOffRedLed      BlinkRedLed        Don't turn OFF  the RED Led while it is bl
Sound Detect Warning         drive      SoundDetection     BlinkRedLed        Blink the Red LED if a sensor detect sound
Environment Brightness Check support    TurnOnRedLed       LightDetection     The Red LED could only be turned on when t
Prevent Redundant Trigger    interfere  WarningThruLED     WarningThruLED     Prevent multiple trigger the warning syste
```

✦ show all

I. Show all types of received Tweets in one command

✦ recipe

I. Prompt the user to create and finalize an APP

```
recipe

Execute cmd recipe

To finalize an APP, you could add one or two service(s) for an APP
(Notice: There is only one relation you could add for two services, but none for one service)
How many services you would like to add?
Please enter 1 or 2:
```

II. An APP could support a single Service or two Services

```
Now, please choose the service you would like to add from below
[SpaceID]          [ThingID]          [Name]             [Input 1]               [Input 2]          [Output]
Group4VSS          MyRPI_5341         TurnOnRedLed       void                    void               void
Group4VSS          MyRPI_5341         TurnOffRedLed      void                    void               void
Group4VSS          MyRPI_5341         GetRedLEDStatus    void                    void               LED_Status
Group4VSS          MyRPI_5341         SoundDetection     PeriodOfTimeInSeconds   void               soundDetected
Group4VSS          MyRPI_5341         LightDetection     PeriodOfTimeInSeconds   void               isBright
Group4VSS          MyRPI_5341         BlinkRedLed        times                   void               void


Enter the "ThingID" and "ServiceName" pair seperated by a space below (ThingID ServiceName):
```

III. A relationship could be added to a APP containing two Services

IV. If user added 2 Services, the order of entering Service 1 and Service 2 would effect the relationship

V. If any matched relationship are found in received Relationship Tweets, it will be shown with the default relationships. In the image below, a Relationship from tweet, “Sound Detect Warning”, is shown.

VI. If the special relationship is chosen by the user, the IDE will convert it to one of the supported type. “Drive” relationship would be set if user chooses “Sound Detect Warning”

```
[Relationships]               [Description]
Sound Detect Warning          (drive) Blink the Red LED if a sensor detect sound
control                       If A THEN B
drive                         USE A TODO B
support                       BEFORE A CHECK ON B
extent                        DO A WHILE DOING B
contest                       Not supported by IDE yet
interfere                     Not supported by IDE yet

Please choose a prefered relationship for these two Services:
Sound Detect Warning
"Sound Detect Warning" belongs to "drive" relationship type
```

VII. Each Service added would prompt the user to set the input(s) and expected output result, at most 2 inputs and 1 output

VIII. The Service API and input / output description would be shown

```
Please set the input and expected output result for Service 1 if required ...

SoundDetection:[PeriodOfTimeInSeconds,int, NULL]:(soundDetected,int, NULL)
Required number of integer Input(s): 1
        Description of Input 1: PeriodOfTimeInSeconds
        Description of Input 2: void


Enter an integer for Input 1:
1
```

IX. The expected output result is used to determine if the Service is successfully executed. For now, we only support “equality” between expected result and actual output of the Service.

X. If the Service does not have any specified input and output, the IDE will set them to default value NULL

```
Please set the input and expected output result for Service 2 if required ...

BlinkRedLed:[times,int, NULL]:(NULL)
Required number of integer Input(s): 1
        Description of Input 1: times
        Description of Input 2: void


Enter an integer for Input 1:
5
BlinkRedLed:[times,int, NULL]:(NULL)
No output, the expected result would be the success of activating this service
```

XI. After all the required settings for the APP, it will show a brief setting information in console

```
Brief APP settings you just made:
[Service 1]                 [Input(s)]                  [Expect Result]
SoundDetection              (1)                         (5)

[Service 2]                 [Input(s)]                  [Expect Result]
BlinkRedLed                 (5)                         Successfull Call

[Relationship Type]         [Description]
drive                       USE A TODO B

Enter an unique name for this APP or "clear" to give up:
```

XII. In the last step, user should give this new APP a name to save it. If the user does not like the setting, the user could type “clear” to give up.

✦ app

I. User could activate the APPs created in recipe command for one time

```
app

Execute cmd app

All the APPs are listed below:
[ID]    [APP Name]              [Service 1]             [Relationship]          [Service 2]
1       fff                     TurnOnRedLed
2       1                       LightDetection
3       aer                     SoundDetection          drive                   BlinkRedLed
Enter the APP "ID" you would like to activate:
3
```

II. Showing the result reply Tweet status for execution each service in the APP

III. If an APP contains a relationship failed, it would show which service cause the APP to fail

```
------------------ACTIVATION--------------------
aer: USE "SoundDetection" TODO "BlinkRedLed"
------------------------------------------------


Reply Tweet info for activated service "SoundDetection"
Status:             Successful
Description:        Successful Service Call
Result:             0


APP "aer" failed because an unseccessfull service call or the replied result is not expected for "SoundDetection"

-------------------FAILED-----------------------
aer: USE "SoundDetection" TODO "BlinkRedLed"
------------------------------------------------
```

Limitations and Possible Features

1. IDE supports at most 2 services and 1 relationship for an APP
2. Expected result predicate only support equality, it would be more powerful to support comparisons such as bigger than, smaller than or not equal
3. Relationship Contest and Interfere are not supported
4. APP could only be activated once, maybe some features for scheduling to activate an APP periodically
5. There could be some relationship between APPs
6. Save, Upload, Stop, Delete is not supported in Application Manager yet
7. Unbounded Services is not supported yet, due to no this kind of Tweet
8. Not support to show a graph for services as vertex, relationships as edges