DEVELOPMENT OF A TASK-ORIENTED CHATBOT APPLICATION FOR MONITORING TAIWAN PHOTON SOURCE FRONT-END SYSTEM

Yu-Zheng Lin*, Jyun-Yan Chuang, I-Ching Sheng, Yu Tsun Cheng,
Chin-Chun Chang, Yi-Chen Yang, Hsin-Pai Hsueh, Chih-Hsien Huang
National Synchrotron Radiation Research Center, Hsinchu, Taiwan

Abstract

In this study, we propose a task-oriented chatbot application as an interactive user interface for monitoring Taiwan Photon Source front-end system. This application can get specific information faster and improve the efficiency of the maintenance engineer's definition of fault information when a fault occurs. The chatbot uses LINE's Message API to provide services, and obtains information and status of the front-end system over EPICS protocol, responding to users' needs, like a virtual assistant. At present, the application can obtain the front-end system information already in operation of Taiwan Photon Source, including x-ray beam position monitor, safety interlock system, front-end valve status, vacuum status, etc. However, in addition to the passive provision of information, this program has a fault warning function, and will actively transmit fault information to relevant personnel when a problem occurs in the safety interlock system.

INTRODUCTION

The chatbot application is a dialogue system. Unlike a GUI system that provides a lot of information, the dialogue system is search-based. When the user asks for information, the dialogue system returns the answer. In this study, we propose a chatbot application for monitoring Taiwan Photon Source(TPS) front-end system. This chatbot will make engineer and scientist get TPS front-end status easier and improve work efficiency.

The chatbot application is based on LINE message platform. LINE is a cross-platform communication software that can be easily used on mobile phones, tablets, and personal computers. LINE provides "LINE Message API" [1] with a complete development solution, allowing developers only need to develop once and users can use this application on different platforms. This app has two main functions, one is to reply to the user query and the other is the active fault alert. The reply function is when the server receives the user query, service will use the rule-based engine to filter out the keyword, then uses "Experimental Physics and Industrial Control System"(EPICS) [2] call the associated EPICS process variable to get status information and finally passes the results back to the user. This service also continuously monitors the status of the system. When a failure is detected, it will actively send messages to the responsible personnel of the system.

* lin.yz@nsrrc.org.tw

CHATBOT ARCHITECTURE

The chatbot architecture is shown in Figure 1. We applied to LINE for a channel to implement this chatbot application, this channel like a friend in a social network, users just add this channel as a friend and can use it. The chatbot application is all made by Python, the LINE message API is used to connect to the LINE channel and use the push and reply methods to make fault alerts and reply to the status results queried by the user.

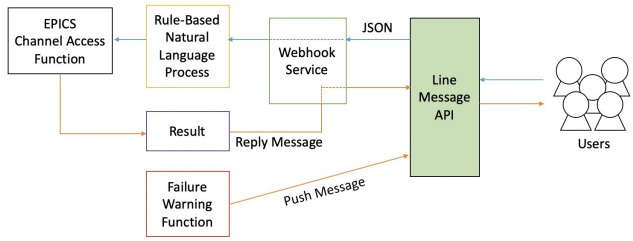

Figure 1: Chatbot architecture.

Reply Function

The Reply function has three parts: Webhook Service, Rule-Based Natural Language Process Engine and EPICS Channel Access Function.

Webhook Service is a HTTP server [3]. To implement this service, python and the flask web framework [4] is used. When a user sends a message to the chatbot, webhook service will receive a JSON file from LINE message API, this JSON file include message information, event source and time. The screenshot of the server operation is shown in the Figure 2.

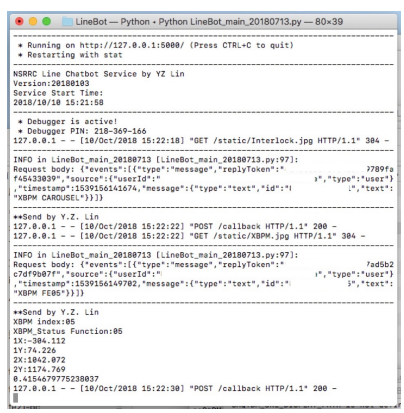

Figure 2: Screen shot of the webhook service operation.

After receiving the message, the Rule-Based Natural Language Process Engine will filter out the keywords in the

message. Then use the EPICS Function callback related values according to different keyword rules. Finally, the value is filled in the formatted string of the result and use reply method from LINE message API send back to the user who sent the query.

Figure 3 is an example of how X-Ray beam position monitor information is returned. The rule-based engine will detect if "XBPM" and "FE" is in the text, and filter out the front-end numbers if the condition is true. After filtering out the TPS front-end numbers, this numbers will be filled in "XBPM_Status" subroutine to get X-Ray beam position value. Finally, fill the value into the formatted result string and use LINE message API push method returns the result to the user.

```
if "XBPM" in strupper and "FE" in strupper:
    str=event.message.text
    FE_number=re.sub("\D","", str)#take out FE number
    print('XBPM index:%s'%FE_number)
    if bool(FE_number)==True:
        try:
            #Use EPICS to get XBPM value
            XBPM_1X, XBPM_1Y, XBPM_2X, XBPM_2Y = XBPM_Status(FE_number)
            #Use LINE message API reply method return the result
            line_bot_api.reply_message(event.reply_token,TextSendMessage\
                                       (text="FE%s XBPM Position:\n"%FE_number+\
                                       "• 1X: %s um\n"%XBPM_1X+\
                                       "• 1Y: %s um\n"%XBPM_1Y+\
                                       "• 2X: %s um\n"%XBPM_2X+\
                                       "• 2Y: %s um"%XBPM_2Y))
        except:
            print("No Index")
            return 0
    else:print("No Index")
    end = time.time()
    elapsed=end-start
    print(elapsed)
    return 0
```

Figure 3: Example of how X-Ray beam position monitor information is returned.

Active Fault Alert Function

Active fault alert function is currently used on the TPS front-end safety interlock system. This function monitor ten TPS front-end safety interlock systems status with EPICS. When the safety interlock system fails, this function will send a warning message to the responsible person using LINE message API push method. The message will include possible causes of failure based on the logic of the safety interlock system.

CHATBOT APPLICATION

The user experience of using this chatbot application is like chatting with a friend, very simple and quite easy to understand. The LINE user just needs scans the QR code of the chatbot and adds the chatbot to the friend to start using the chatbot application.

The user only needs to enter the text to be queried, if this string of text matches the rule-based engine definition and the chatbot will return the result. Figure 4 is an example of the user querying X-Ray beam position monitor status and TPS front-end valve, the figure shows that the value of X-Ray beam position monitor and TPS front-end valve status is returned.

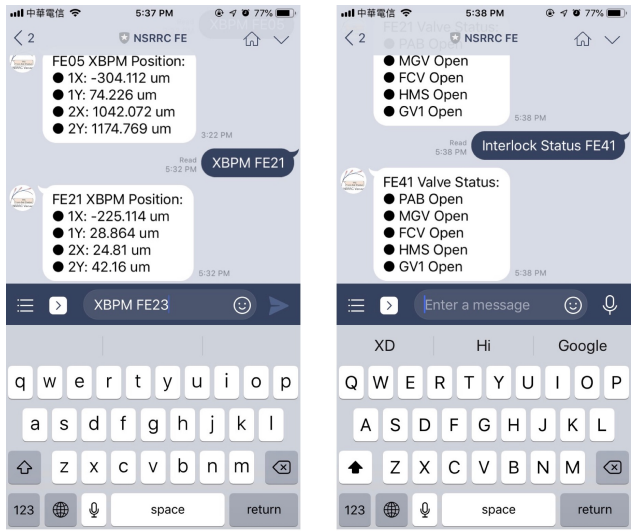

Figure 4: Screen shot of the Chatbot application: (A) Query X-Ray beam position monitor information, (B) Query TPS front-end valve status.

CONCLUSION

The chatbot application used to query and provide fault warning information for the front-end system of Taiwan Photon Source has completed the prototype. The application can obtain the front-end system information already in operation of Taiwan Photon Source, including x-ray beam position monitor, safety interlock system, front-end valve status, vacuum status, etc. The future will increase the amount of information that can be provided and enhance the user experience. We also consider providing services on different platforms, such as Facebook, WhatsApp, etc., so that international users of Taiwan Photon Source can use this application more conveniently.

REFERENCES

[1] Messaging API - LINE Developers, https://developers.line.me/en/docs/messaging-api/overview/

[2] EPICS - Experimental Physics and Industrial Control System, https://epics.anl.gov/

[3] Web hooks to revolutionize the web, http://progrium.com/blog/2007/05/03/web-hooks-to-revolutionize-the-web/

[4] Welcome to Flask — Flask 1.0.2 documentation – Pocoo http://flask.pocoo.org/docs/1.0/