Accessing and Decoding METAR and TAF on Mobile Devices

using Web Services

Kasikrit Damkliang [+], Ekkarat Muangdong and Thiraphat Prakobkarn

Information and Communication Technology Programme

Faculty of Science, Prince of Songkla University

Hat Yai, Songkhla, Thailand.

Abstract. This paper presents web services for accessing and decoding high accuracy weather information in forms of METAR (Meteorological Aviation Report) and TAF (Terminal Aerodrome Forecast) on mobile devices. Our system consists of two co-operated components. The first component concerns web services providing raw and decoded METAR and TAF information as a service provider. The rest is a service requester which our mobile application can access via SOAP. Smartphone users can also access services via Wi-Fi and cellular networks.

Keywords: web service, access, decode, METAR, TAF.

1. Introduction

Nowadays, weather forecasting becomes increasingly important in our daily lives. There are many applications on pocket PCs or Smartphones providing weather information for users. However, these applications are not free services and have platform limitation problems [1][2]. We present a free application and services for reporting weather information especially for aviation industry.

METAR (Meteorological Aviation Report) and TAF (Terminal Aerodrome Forecast) report are generated by the World Meteorological Organization [3] in cooperation with ICAO (International Civil Aviation Organization) [4]. METAR and TAF are high accuracy weather news services used as input for check-up routines in order to prepare flight plans. Nowadays, staffs in the aviation field, meteorologists and pilots access them via an automatic telephone system or the Internet using a PC. Nevertheless, pilots always access them via a radio channel on the airplane's cockpit during flight operation [5][6].

Both data, METAR and TAF, is formatted in a codified observation message. Therefore, it is necessary to decode it into an understandable message for general users. In this paper, we develop and deploy an application for accessing and decoding METAR and TAF on mobile devices using web services. Our application offers an additional access channel for the information. It is convenient and quick for pilots or meteorologists. Especially, pilots can access this information anytime anywhere for decision making for their flight plan.

2. Background

2.1. METAR and TAF

[+] Corresponding author. Tel.: + 6674288697; fax: 6674288697.
E-mail address: kasikrit@psu.ac.th

2010/10/08 06:00 VTBS 080600Z 21006KT
9999 FEW020CB SCT035 BKN300 33/24
Q1009 CB NW NOSIG

Fig 1. Example of Suvarnabhumi Airport METAR

METAR is a current aerodrome routine meteorological report and created by Aviation Routine Weather Observation at thousands of primarily airports around the world every half an hour, 24 hours a day. Raw METAR is the most popular format in the world for the transmission of weather data. It is highly standardized through ICAO, which allows it to be understood throughout most of the world [7].

A typical METAR report contains data for the temperature, dew point, wind speed and direction, precipitation cloud cover and heights, visibility and barometric pressure. A METAR report may also consist of information on precipitation amount, lightning, and other information that would be interested to pilots or meteorologists such as a pilot report and runway visual range (RVR), etc.

TAF is a format for reporting weather forecast information. TAFs apply to a five statute mile radius from the center of the airport. Commonly, TAFs can apply to a 9 and 12 hour forecast. TAFs for some major airports cover 30 hours periods. The date time group reflects the new 30 hour periods in Coordinated Universal Time (UTC). TAFs complement and use similar encoding to METAR reports. [8]. The complement to METAR reports, reporting forecast weather rather than current weather, are TAFs.

Figure 1 shows a simple example of METAR and Table 1 is the decoded METAR for Suvarnabhumi Airport of Thailand.

Code	Meaning
VTBS	Location: In ICAO code
080600Z	Time: 06:00 UTC on 8th of the month
21006KT	Wind: True direction 210 degree, 6 knots
9999	Visibility: 10 km or more
FEW020CB	Clouds: A few, at 2000 feet above aerodrome level, cumulonimbus
SCT035	Clouds: Scattered , at 3500 feet above aerodrome level
BKN300	Clouds: Broken sky , at 30000 feet above aerodrome level
33	Temperature: 33 degrees Celsius
24	Dew point: 24 degrees Celsius
Q1009	QNH (Sea-level pressure): 1009 hPa
NO SIG	No significant changes expected in the near future

Table 1. Decoded information of Suvarnabhumi Airport METAR

2.2. Web Service

A web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine process-able format specifically, Web Service Definition Language (WSDL). Other systems interact with the Web service in a manner prescribed by its description using Simple Object Access Protocol (SOAP) messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. This project is designed by applying some parts of Service-oriented Architecture (SOA) framework for implementing web services.

2.3. Java Runtime Environment

In this project, we use Connected Limited Device Configuration (CLDC) as a virtual machine on Smartphones and implement Web Services APIs (WSA) 1.0 for providing standard access from Java 2 Platform, Micro Edition (J2ME) devices to web services [9]. CLDC is a fundamental part of the J2ME architecture.

The J2ME technology is delivered in API bundles called configurations, profiles, and optional package. It defines the base set of API and a virtual machine for resource-constrained devices like mobile phones and personal digital assistants.

CLDC is designed to bring the many advantages of the Java platform to network-connected devices that have limited processing power, memory, and graphical capability, such as mobile phones, low-end personal organizers, and machine-to-machine equipment.

Java Platform, Enterprise Edition (Java EE) is the industrial standard platform for building enterprise-class applications coded in the Java programming language. In this project, we use Java EE platform in managing and deploying application web services.

3. System Design

3.1. System Architecture

Our system consists of two co-operated components as shown in Figure 2. The first component concerns web services providing raw and decoded METAR and TAF information as a service provider or server-side. The second component is a client-side such as a service requester on which our mobile application is installed a Smartphone and may be a browser accessing via SOAP. The Smartphone user can also access services via Wi-Fi and cellular networks.

Figure 3 explains data flow in the system. First, the user inputs an airport ICAO code on a mobile device in Process 1.0. Then, in Process 2.0, a mobile application calls and sends the airport ICAO code to a web service on the server. The application web service receives the airport ICAO code and stores it into variables in Process 3. 0. Next step, the service makes a request for accessing METAR/TAF information to the dataprovider, Bureau of Aeronautical Meteorology, Thailand. The data provider returns METAR/TAF information in a HTML tag to Process 4.0. The information in a HTML tag is then sent to Process 5.0 for decoding. Finally, the decoded information is returned and presented on a mobile device as shown in Process 6.0 and Process 7.0 consecutively.

The services overview is shown in Figure 4. Calling Web Service is a control center coordinating between requesters and invocating, getting and decoding METAR/TAF, and viewing station or airport information.

Fig 2. System Architecture of Application Web Services

3.2. Service and Class Design

Getting and Viewing METAR/TAF Sequence Diagram in Figure 4 describes that the application input is an ICAO airport code works as a key for accessing METAR and TAF from Bureau of Aeronautical Meteorology, Thailand [10]. In Figure 5 Decoding METAR Sequence Diagram shows our services can decode METAR and TAF formats into readable messages for general users in

Fig 3. Data Flow Diagram of the System for Accessing Aviation Weather Using Web Services

the case that the user does not know an ICAO code of a required airport. The Viewing Station Service returns station details such as airport name, ICAO code, location coordinate and country code. We organize this information into an XML format as shown in Figure 8.

4. Implementation and Deployment

4.1. Implementation

We apply Java programming and Sun Java Wireless Toolkit 2.5.2_01 for CLDL in order to develop a mobile application in client-side. On the service side, we apply Java EE 6 SDK and Glass Fish Server 3.0.1 for hosting the web services. We also use NetBeans IDE 9.6.1.
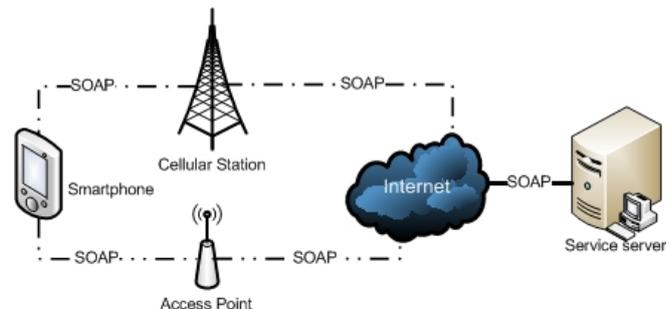
We implement HTTP Connection statements for getting the ICAO code on the client side and using it as a key to query for METAR/TAF data from Bureau of Aeronautical Meteorology and then decode it on the server side, an example is shown in Figure 7. Next step, decoded METAR/TAF data is delivered to client, in this case, a Smartphone. In addition, we use SAX Parser to look up in the station XML file in Figure 8 for querying for station details.
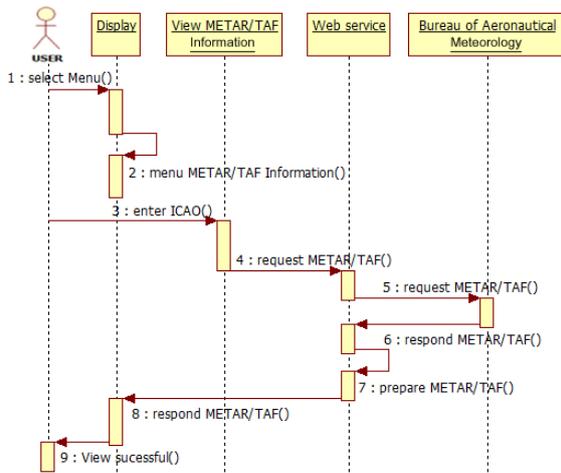

Fig 4. Getting and Viewing METAR/TAF Sequence Diagram

Fig 5. Decoding METAR Sequence Diagram

```
<?xml version="1.0" encoding="utf-8"?>
<station>
    <country name="THAILAND" >
        <airport>SUVARNABHUMI/SATTAHIP,VTBS,BKK,4845
9,13 41N,100 45E,2,7 TH</airport>
        <airport>BINTULU/KALIMANT,WBGB,-,96441,03
12N,113 01E,5,7 MY</airport> </country> </station>
```

Fig 8. Example of Station or Airport Details Organized into XML

4.2. Deployment

We deploy our application web services on Sun Glass Fish, an application server. The WSDL endpoint is:

http://ict.sci.psu. ac.th:8080/AerometMobileWS /Aeromet MobileWSService?WSDL

We test application web services via a browser with URL:

http://ict.sci.psu.ac.th:8080/AerometMobileWS/Ae rometMobileWSService?tester

Then, the user can download the client application for installing on a mobile device from:

http://ict.sci.psu.ac.th:8080/AerometMobileWS

5. Experimental Results

We test our client mobile application on a Samsung Candy S3650. The device supports Java application with Java Mobile Information Device Profile (MIDP) 2.0.
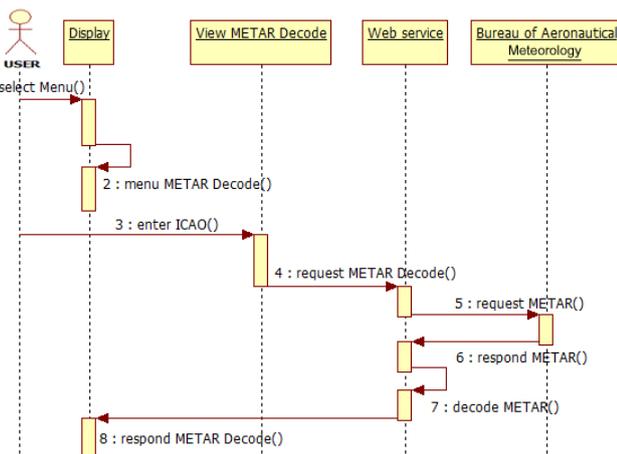
Figure 9 (a) shows mobile application home screen on the smartphone consisting of three choices, the first is METAR/TAF Information for getting aviation weather data, and the rest is METAR and TAF Decode for decoding METAR and TAF from a required airport in ICAO code consecutively. Figure 9 (b) shows METAR information of VTSS at Hatyai Airport, Songkhla, Thailand.

Figure 10 (a) shows input screen on the mobile for accepting ICAO station code of any airport. Figure 10 (b) shows decoded METAR of VTSS at Hatyai Airport, Songkhla, Thailand.

The application is also tested by meteorologists at Aero Meteorological Department, East-Shore Southern Meteorological Center of Thailand. They are satisfied by the application and also verify the results.

```
Proxy proxy = new Proxy(Proxy.Type.HTTP, new
InetSocketAddress("cache.sci.psu.ac.th",8080));
URL url = new URL("http://www.aeromet.tmd.go.th /_pages /opmet/
detail/data.asp?MODE=QS&AEROS="+AEROS+"&REQDATE=&
HIST=1&METAR=1&TAFL=1");
URLConnection connection = url.openConnection(proxy);
```

Fig 7. Code fragment for querying for METAR and TAF data from Bureau of Aeronautical Meteorology on the server side

6. Conclusions

Our application web services give satisfied and verified experimental results. The application web services access, decode and present accuracy

weather information. The system operates correctly in both the mobile application on the client side and the web services on the server side. We only experiment at only example airports in Thailand. Technically, the system can also query aviation weather information of station or airport data all over the world [10]. Therefore, the system adds an additional access channel for those who are interested. Especially, pilots can access this information anytime anywhere for decision making for their flight plan.

In future work, the system would display the map of required places and weather satellite photographs for pilots or meteorologists. In case of general users, the system would give brief advices about travelling to airport or nearby places.

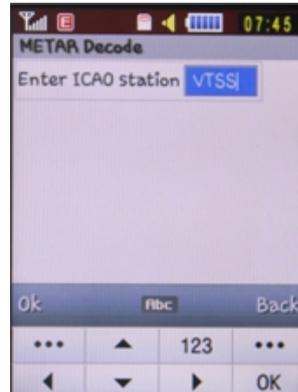Fig 9. (a) Mobile applicaton home screen

Fig 9. (b) METAR of VTSS
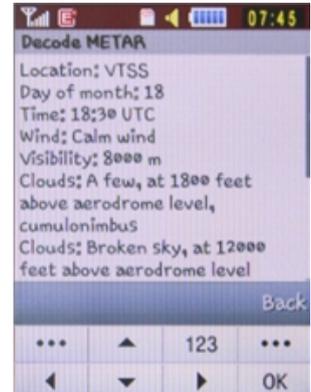
Fig 10. (a) Input screen for accepting ICAO station

Fig 10. (b) Decoded METAR of VTSS

7. Acknowledgement

8. References

[1]   Pocket METAR, http://www.aviationreviews.com/ pocketme.html, 2010.

[2]   AeroWeather for iPhone/iPod touch/iPad, http://www. aeroweather.ch/, 2010.

[3]   World Meteorological Organization, http://www.wmo .int/pages/index_en.html, 2010.

[4]   ICAO | International Civil Aviation Organization , http ://www.icao.int/, 2010.

[5]   Zhengwei Guo, Jinan Zhang , "An approach for METAR report decode based on finite automata" Proc. IEEE Granular Computing, 2009, GRC '09, IEEE Press, 17-19 Aug. 2009, pp. 188-191. doi: 10.1109/GRC.2009.5255131.

[6]   Galati, G., Dalmasso, I., Pavan, G., Brogi, G., "Fog Detection Using Airport Radar", Proc. IEEE Radar Symposium, 2006. IRS 2006, IEEE Press, 24 – 26 May 2006, pp. 1-4, doi: 10.1109/IRS.2006.4338037.

[7]   METAR Data Access, http://weather.noaa.gov/weather /metar.shtml, 2010.

[8]   TAF Data Access, http://weather.noaa.gov/weather/ taf.shtml, 2010.

[9]   J2ME Web Services APIs (WSA), JSR 172, http://java .sun.com/products/wsa/, 2010.

[10] Bureau of Aeronautical Meteorology, Thailand, http://www.aeromet.tmd.go.th, 2010.