DOI 10.5162/ettc2022/9.4

TABAI. Test Assistant Based on Artificial Intelligence

Francisca Coll Herrero[1], Pedro Rubio Alvarez[1]
[1] *Airbus Defense and Space, Av John Lennon s/n, Madrid, Spain*
Francisca.Coll@airbus.com
Pedro.R.Rubio@airbus.com

Abstract:
An Artificial intelligence (AI) is the simulation of human intelligence processes by machines, especially computer systems.
AI include expert systems, natural language processing, speech recognition, machine vision, pattern recognition, etc.
Artificial intelligence has been used in Test for several years specifically in the field of Computer Vision - for example in safe separation (store trajectory calculation), aerial delivery (paratrooper's trajectory calculation), and refuelling manoeuvres (approach speed calculation).
Also other projects where a prototype in which we have already developed or there is an interest in its study and evaluation, such as: pattern recognition based on wavelets for automatic manoeuvre detection (BMAD), real time object detection (basket detection, relative position, approach speed…), measuring pilots workload (eyes, head and hand movement) and FTI validation (Parameters anomaly detection, bad calibrated sensors, failure prediction).
This paper will describe a project called TABAI which consists of a Test assistant based on artificial intelligence.
TABAI is an assistant like Alexa or Siri to help in the Test activities. Using TABAI it will be possible to access to all the Test information that currently is available in a database or in documents or stored in data files.
The potential of TABAI is enormous, from obtaining the maximum value of a flight parameter to become the entry point for many test tools.

Key words: TABAI: Test Assistant Based on AI, Chatbot, AI: Artificial Intelligence, NPL: Natural Language Processing, LM: language Model, FTPR: Flight Test Program Requirements, TPKEY: Test Point Key

1 Introduction

The Airbus Defense and Space organization has grown a lot in recent years, so it is increasingly difficult to access required information quickly and efficiently.

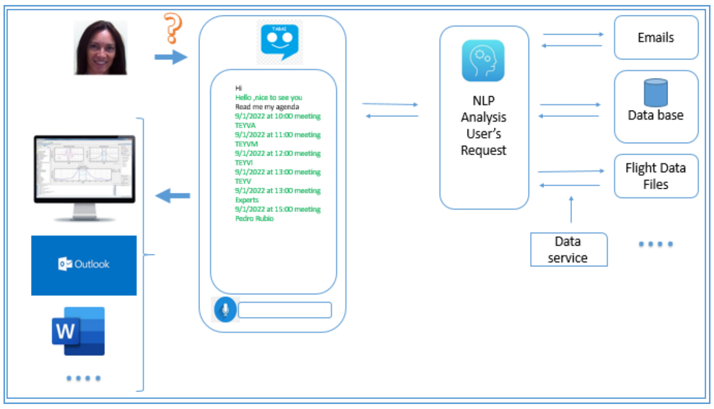
This is the reason why **TABAI has been envisaged to be develop.**

TABAI is a Text Assistant based on Artificial Intelligence with the objective to help Test activities such as plotting Test parameters corresponding to a specific Test, to check the status of a program, to find documents using key-words, etcetera.

The interface of **TABAI** is a Chatbot that interprets the user's requirements using Natural Language Processing,

Fuzzy String searching that interprets the request and a server who finds the information and sends it to the Chatbot.

In the Figure 1 you can see the workflow of **TABAI.**



Hi
Hello ,nice to see you
Read me my agenda
9/1/2022 at 10:00 meeting TEYVA
9/1/2022 at 11:00 meeting TEYVM
9/1/2022 at 12:00 meeting TEYVI
9/1/2022 at 13:00 meeting TEYV
9/1/2022 at 13:00 meeting Experts
9/1/2022 at 15:00 meeting Pedro Rubio

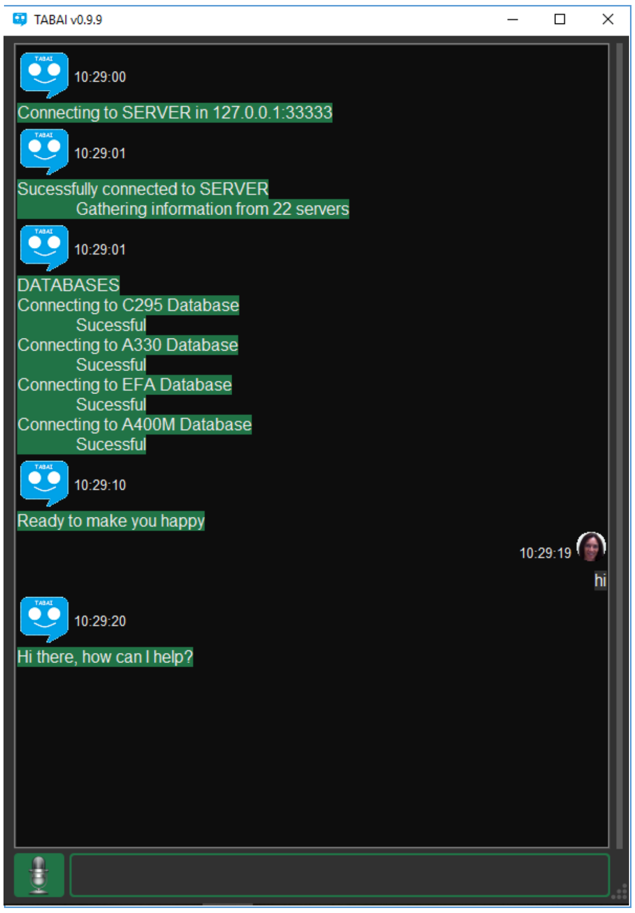
Figure 1.TABAI Workflow

2 TABAI components

TABAI consists of a Chatbot-like interface with a multiple of utilities such as accessing Test information contained in databases, Test data files, Test document files, Test video files, running analysis tools , read out emails and finally a server that finds and provides the information required by the user.

2.1 TABAI tool

The interface is a Chatbot developed in Python [Ref.1].


TABAI v0.9.9

10:29:00
Connecting to SERVER in 127.0.0.1:33333

10:29:01
Sucessfully connected to SERVER
Gathering information from 22 servers

10:29:01
DATABASES
Connecting to C295 Database
Sucessful
Connecting to A330 Database
Sucessful
Connecting to EFA Database
Sucessful
Connecting to A400M Database
Sucessful

10:29:10
Ready to make you happy

10:29:19
hi

10:29:20
Hi there, how can I help?

Figure 2. Interface

TABAI tool is composed by:

- **Intents.json** :JSON File that lists different tags that correspond to the different chat's inputs and the corresponding chat´s answers used in the Chatbot.
- **Chatbot_engine.py**: A python class in charge of interpreting user requests based on NLP and fuzzy string machine techniques [Ref.2]. That request is compared with the list contained in Intents.json and the most similar one is selected.
- **Chatbot_DB** . Python package to access to database.
- metaClient.py : Python package to access to file flight via server.

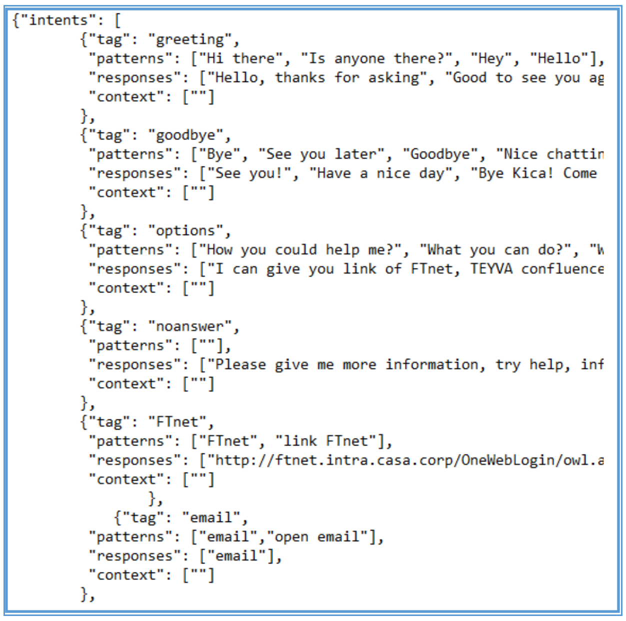

```
{"intents": [
        {"tag": "greeting",
         "patterns": ["Hi there", "Is anyone there?", "Hey", "Hello"],
         "responses": ["Hello, thanks for asking", "Good to see you ag
         "context": [""]
        },
        {"tag": "goodbye",
         "patterns": ["Bye", "See you later", "Goodbye", "Nice chattin
         "responses": ["See you!", "Have a nice day", "Bye Kica! Come
         "context": [""]
        },
        {"tag": "options",
         "patterns": ["How you could help me?", "What you can do?", "W
         "responses": ["I can give you link of FTnet, TEYVA confluence
         "context": [""]
        },
        {"tag": "noanswer",
         "patterns": [""],
         "responses": ["Please give me more information, try help, inf
         "context": [""]
        },
        {"tag": "FTnet",
         "patterns": ["FTnet", "link FTnet"],
         "responses": ["http://ftnet.intra.casa.corp/OneWebLogin/owl.a
         "context": [""]
                },
            {"tag": "email",
         "patterns": ["email","open email"],
         "responses": ["email"],
         "context": [""]
        },
```

Figure 3. Intents.json

- **TABAI_GUI.py** : Is the graphics interface that interacts with the user.
- **Metaserver** : A program written in C++ that serves to the client the user's requests via socket.
- **metaClient.py** : A python class that sends/receives the user's request to the Metaserver.
- **TABAI.py** : Main program in Python with the functions of user requirements interpretation using the chatbot_engine class ,execution of the corresponding actions using metaClient class and finally sending it in the TABAI_GUI.

2.2 Chatbot engine

A Chatbot is a computer program that simulates human conversation through text or text-to-speech.

A critical part of Chatbot implementation is selecting the right natural language processing engine (NLP).

In the first version of TABAI, a Keras sequential neural network model was used as NLP. This model was trained using the list of sentences included in the Intents.json file.

Afterwards a Fuzzy String Comparison based on:

- Levenshtein distance
- Sort alphabetically words
- Removing blank spaces between words in sentences

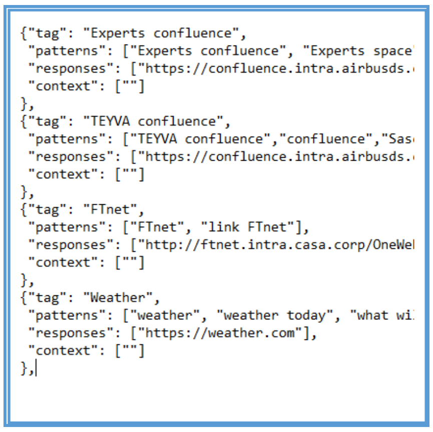
was tested

Finally, it was decided to use **Fuzzy String Comparison** since it worked much better for this use case.

2.3 Access to links

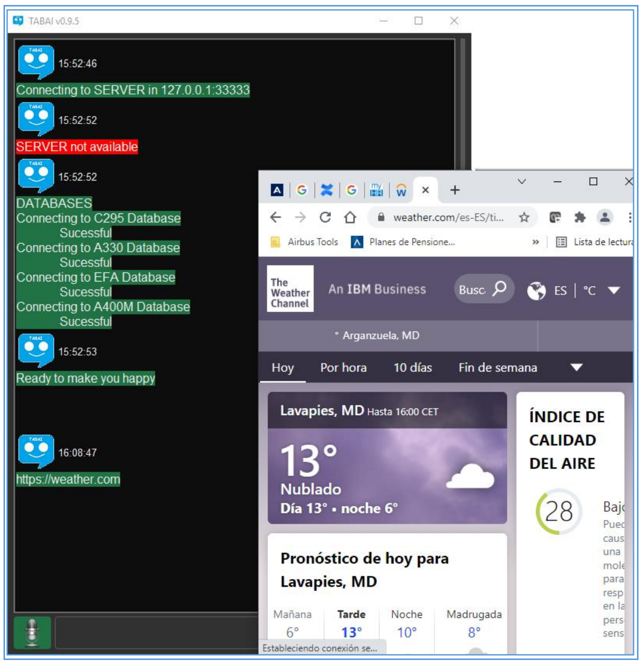
This is the simplest function of **TABAI**, the links are directly included as a response in the Intents.json

In the Figure 4 you can see how are introduced the links in the json file and in the Figure 5 how use this feature in **TABAI** interface.



```
{"tag": "Experts confluence",
 "patterns": ["Experts confluence", "Experts space
 "responses": ["https://confluence.intra.airbusds.
 "context": [""]
},
{"tag": "TEYVA confluence",
 "patterns": ["TEYVA confluence","confluence","Sas
 "responses": ["https://confluence.intra.airbusds.
 "context": [""]
},
{"tag": "FTnet",
 "patterns": ["FTnet", "link FTnet"],
 "responses": ["http://ftnet.intra.casa.corp/OneWel
 "context": [""]
},
{"tag": "Weather",
 "patterns": ["weather", "weather today", "what wi
 "responses": ["https://weather.com"],
 "context": [""]
},
```

Figure 4. TABAI links



```
TABAI v0.9.5
15:52:46
Connecting to SERVER in 127.0.0.1:33333
15:52:52
SERVER not available
15:52:52
DATABASES
Connecting to C295 Database
          Sucessful
Connecting to A330 Database
          Sucessful
Connecting to EFA Database
          Sucessful
Connecting to A400M Database
          Sucessful
15:52:53
Ready to make you happy
16:08:47
https://weather.com
```

Figure 5. TABAI link

2.4 Access to Test Files information

During the last 30 years, several formats for storing Test Files have been used, depending on the current technology and the upcoming requirements. As an example, 30 years ago the number of available parameters were around hundreds and nowadays we manage hundreds of thousands, therefore the format to store these parameters has completely changed.

As we realized that the format of the Test Files change, and will change in the future, we decided to develop a network based unified protocol to access these data (FxS). In this approach, the final applications (plotting tools, analysis tools, any application that need Test Data) will remain unchanged, regardless the format we used.

For each format and new format, an FxS Server is developed that is able to serve data to any client compatible with this protocol.

DOI 10.5162/ettc2022/9.4

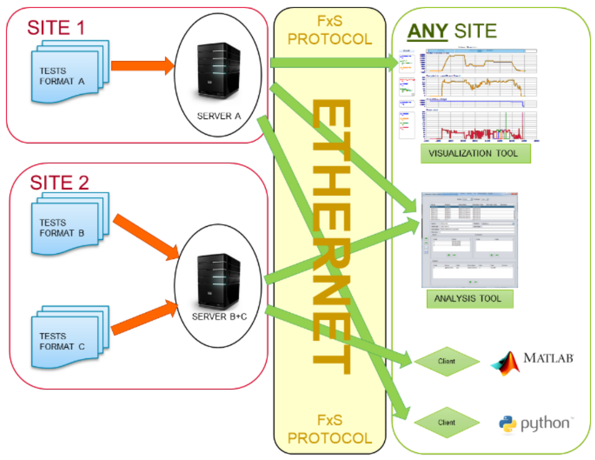


Figure 6. FxS Schematics

We used to handle tens of aircrafts across its lifecycle of testing, and even for a long program the format of the Test File could change.

In such complex scenery, sometimes is difficult to find out which FxS server is capable to give data for a specific aircraft/test. The solution is, the so called, **Metaserver**.

This Metaserver gathers the information of all the FxS running in our ecosystem (currently more than 40, and increasing…), basically gathers two main information:

- Aircrafts available in each server.
- Tests available per each aircraft.

This Metaserver uses a simple UDP network transactions based in XML for the queries and responses. Other ways of communication (json, REST API, and some other) will be evaluate in the future.

For each query, the matching is not an exact match, but a fuzzy string comparison based on the Levenshtein distance.

The Levenshtein distance is a string metric for measuring the difference between two sequences.

Currently, the queries available are:

- **AIRCRAFT**
  - **Query**: pattern of an aircraft.
  - **Response**: aircrafts available in any server that best *matches* the pattern.
- **TEST**
  - **Query:** pattern of an aircraft and pattern of a test.
  - **Response:** Tests available in any server that best *matches* the pattern. For each single test the Metaserver sends all the needed information to connect with the FxS server that really has the test file. If there are some, the Metaserver choose the less loaded.

2.5 Access to Data Base

Some of the information required by the user implies accessing the Test databases. Test have several oracle databases for the different aircraft models.

Each database contains tables with information of flights, programs, tests point, status of the program, etcetera.

Using **TABAI** you can ask for the programs of an aircraft model or even the status of a specific program.

3 User interaction with TABAI

The user can interact with the Chatbox using text or using the voice. **TABAI** interprets user's requirements using Natural Language Processing techniques and also, in the second case, using Speech Recognition Techniques.

In neither case the user is required to write or say a specific phrase and in a specific order, but rather **TABAI** will interpret what the user requires from a list of options.

info C295

Figure 7.User interface

3.1 Speech Recognition Technique

There are a lot of very good voice recognition applications in our devices, in mobile phones Siri, Alexa, in cars, in google meet etc. and it is clear that there is no intention to compete with them. But we decided to develop our own language model with a reduced dictionary in **TABAI** since the Google voice recognition library, apart from not being free, required an internet connection and this was a constraint for our application.

Three models are used in speech recognition to do the match between the audio and the combination of words: the acoustic model, phonetic dictionary and the language model.

CMUsphinx toolkit was used to create the three **TABAI** models. [Ref.3]

DOI 10.5162/ettc2022/9.4

The last step is to generate the acoustic model and to train it to enhance the accuracy of the speech recognition [Ref.5].

4 Uses cases

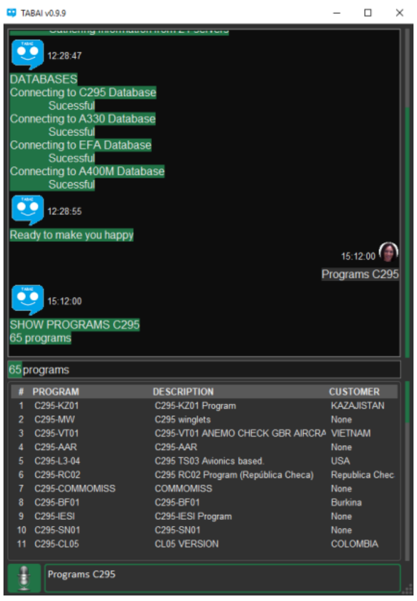
As mentioned in the abstract, the potential of **TABAI** is enormous, in this chapter, some of the use cases that are currently implemented in **TABAI** are described.

4.1 Programs info

The list of programs corresponding to an aircraft model can be obtained using text or the voice.

There are different options, for example writing or saying : programs C295, show C295, programs MRTT, programs efa, show programs C295....

Figure 12 shows the output of **TABAI** in the case that the user selects the C295 programs.



```
12:28:47
DATABASES
Connecting to C295 Database
        Sucessful
Connecting to A330 Database
        Sucessful
Connecting to EFA Database
        Sucessful
Connecting to A400M Database
        Sucessful
12:28:55
Ready to make you happy
                                        15:12:00
                                   Programs C295
15:12:00
SHOW PROGRAMS C295
65 programs
```

65 programs

#	PROGRAM	DESCRIPTION	CUSTOMER
1	C295-KZ01	C295-KZ01 Program	KAZAJISTAN
2	C295-MW	C295 winglets	None
3	C295-VT01	C295-VT01 ANEMO CHECK GBR AIRCRA	VIETNAM
4	C295-AAR	C295-AAR	None
5	C295-L3-04	C295 TS03 Avionics based.	USA
6	C295-RC02	C295 RC02 Program (República Checa)	Republica Chec
7	C295-COMMOMISS	COMMOMISS	None
8	C295-BF01	C295-BF01	Burkina
9	C295-IESI	C295-IESI Program	None
10	C295-SN01	C295-SN01	None
11	C295-CL05	CL05 VERSION	COLOMBIA

Programs C295

Figure 8. Programs C295

TABAI provides the total number of FTPRs corresponding to this program (in this example 32), the global TPKEY status of the program as a percentage of closed, performed, open and pending TPKEYS.

TABAI also shows the list of FTPR with theirs corresponding reference codes, descriptions and total number of TPKEYS. The colour indicates the status that predominates in the TPKEYS.

4.2 Program status

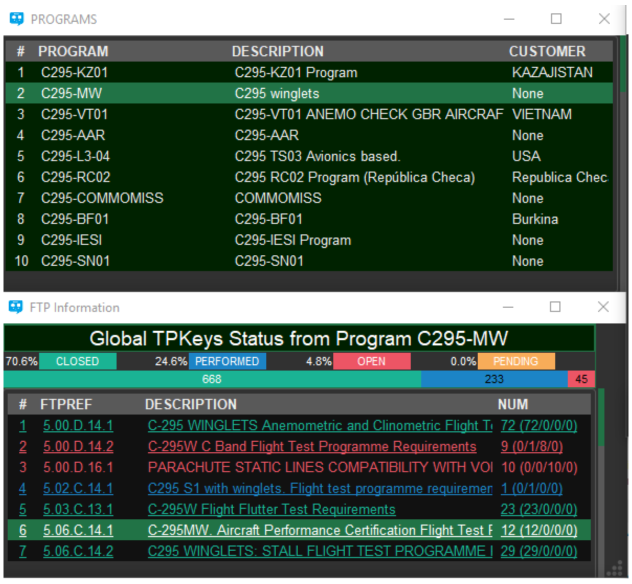
In a Flight Test Program, a list of requirements must be verified and validated. All these requirements are compiled in Flight Test Programs Requirements documents (FTPR).

The list of the requirements for each FTPR are translated into a list of Flight Test Points (TPKEY).

All this information is digitalized and stored in a table of a Flight Test database.

The status of each TPKEY can be OPEN (not flown), PERFORMED (flown but not analysed), CLOSED (flown and validated), and PENDING (flown but not validated).

A summary of the status of the program is obtained clicking on a program in **TABAI** interface.



#	PROGRAM	DESCRIPTION	CUSTOMER
1	C295-KZ01	C295-KZ01 Program	KAZAJISTAN
2	C295-MW	C295 winglets	None
3	C295-VT01	C295-VT01 ANEMO CHECK GBR AIRCRAF	VIETNAM
4	C295-AAR	C295-AAR	None
5	C295-L3-04	C295 TS03 Avionics based.	USA
6	C295-RC02	C295 RC02 Program (República Checa)	Republica Chec
7	C295-COMMOMISS	COMMOMISS	None
8	C295-BF01	C295-BF01	Burkina
9	C295-IESI	C295-IESI Program	None
10	C295-SN01	C295-SN01	None

Global TPKeys Status from Program C295-MW

70.6% CLOSED 24.6% PERFORMED 4.8% OPEN 0.0% PENDING

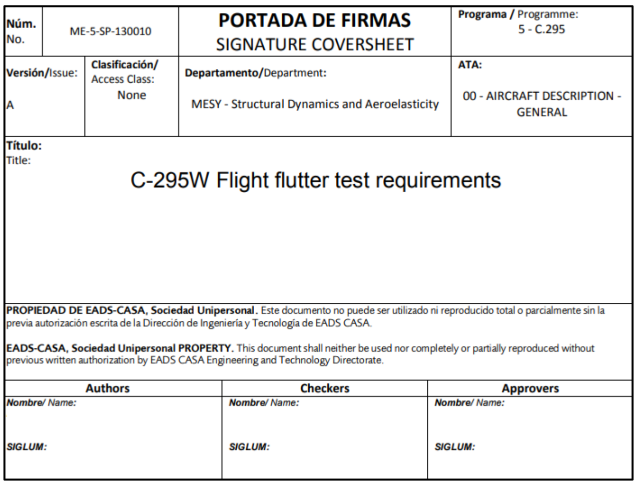
668 233 45

#	FTPREF	DESCRIPTION	NUM
1	5.00.D.14.1	C-295 WINGLETS Anemometric and Clinometric Flight T	72 (72/0/0/0)
2	5.00.D.14.2	C-295W C Band Flight Test Programme Requirements	9 (0/1/8/0)
3	5.00.D.16.1	PARACHUTE STATIC LINES COMPATIBILITY WITH VOI	10 (0/0/10/0)
4	5.02.C.14.1	C295 S1 with winglets. Flight test programme requiremen	1 (0/1/0/0)
5	5.03.C.13.1	C-295W Flight Flutter Test Requirements	23 (23/0/0/0)
6	5.06.C.14.1	C-295MW. Aircraft Performance Certification Flight Test F	12 (12/0/0/0)
7	5.06.C.14.2	C295 WINGLETS: STALL FLIGHT TEST PROGRAMME I	29 (29/0/0/0)

Figure 9. Status program C295 MW

This information could have been accessed directly by writing or saying: status program MW.

The correspondent pdf document can be accessed by double clicking on each FTPREF.



Núm. No.	ME-5-SP-130010	PORTADA DE FIRMAS SIGNATURE COVERSHEET	Programa / Programme: 5 - C.295
Versión/Issue: A	Clasificación/ Access Class: None	Departamento/Department: MESY - Structural Dynamics and Aeroelasticity	ATA: 00 - AIRCRAFT DESCRIPTION - GENERAL

Título:
Title:

C-295W Flight flutter test requirements

PROPIEDAD DE EADS-CASA, Sociedad Unipersonal. Este documento no puede ser utilizado ni reproducido total o parcialmente sin la previa autorización escrita de la Dirección de Ingeniería y Tecnología de EADS CASA.

EADS-CASA, Sociedad Unipersonal PROPERTY. This document shall neither be used nor completely or partially reproduced without previous written authorization by EADS CASA Engineering and Technology Directorate.

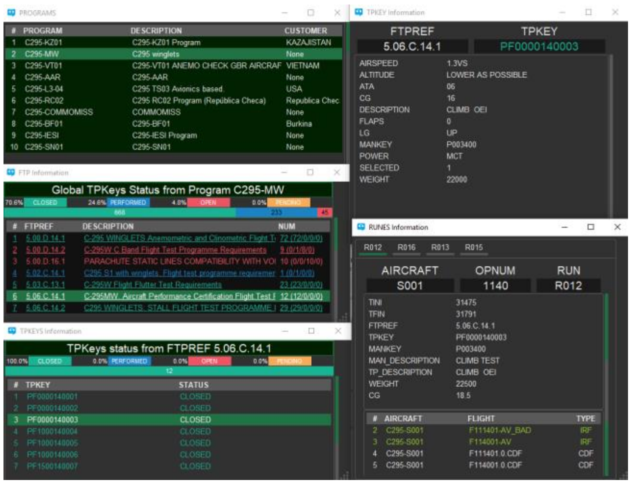
Authors	Checkers	Approvers
Nombre/ Name:	*Nombre/ Name:*	*Nombre/ Name:*
SIGLUM:	*SIGLUM:*	*SIGLUM:*

Figure 10. FTPR document

By clicking on each FTPR, a new window appears containing the list of TPKEYS with its status.

By clicking a TPKEY a window appears showing the information about it.

In case the status of this TPKEY is performed or closed, a second window appears containing the flight and time slices that closed this TPKEY distributed in folders. At the bottom of the window, the list of flight files related to the TPKEY are also shown.



PROGRAMS

#	PROGRAM	DESCRIPTION	CUSTOMER
1	C295-KZ01	C295-KZ01 Program	KAZAJISTAN
2	C295-MW	C295 winglets	None
3	C295-VT01	C295-VT01 ANEMO CHECK GBR AIRCRAF	VIETNAM
4	C295-AAR	C295-AAR	None
5	C295-L3-04	C295 TS03 Avionics based.	USA
6	C295-RC02	C295 RC02 Program (República Checa)	Republica Chec
7	C295-COMMOMISS	COMMOMISS	None
8	C295-BF01	C295-BF01	Burkina
9	C295-IESI	C295-IESI Program	None
10	C295-SN01	C295-SN01	None

FTP Information

Global TPKeys Status from Program C295-MW

70.6% CLOSED 24.6% PERFORMED 4.8% OPEN 0.0% PENDING

668 233 45

#	FTPREF	DESCRIPTION	NUM
1	5.00.D.14.1	C-295 WINGLETS Anemometric and Clinometric Flight T	72 (72/0/0/0)
2	5.00.D.14.2	C-295W C Band Flight Test Programme Requirements	9 (0/1/8/0)
3	5.00.D.16.1	PARACHUTE STATIC LINES COMPATIBILITY WITH VOI	10 (0/0/10/0)
4	5.02.C.14.1	C295 S1 with winglets. Flight test programme requiremer	1 (0/1/0/0)
5	5.03.C.13.1	C-295W Flight Flutter Test Requirements	23 (23/0/0/0)
6	5.06.C.14.1	C-295MW. Aircraft Performance Certification Flight Test I	12 (12/0/0/0)
7	5.06.C.14.2	C295 WINGLETS. STALL FLIGHT TEST PROGRAMME I	29 (29/0/0/0)

TPKEYS Information

TPKeys status from FTPREF 5.06.C.14.1

100.0% CLOSED 0.0% PERFORMED 0.0% OPEN 0.0% PENDING

12

#	TPKEY	STATUS
1	PF0000140001	CLOSED
2	PF0000140002	CLOSED
3	PF0000140003	CLOSED
4	PF1000140004	CLOSED
5	PF1000140005	CLOSED
6	PF1000140006	CLOSED
7	PF1500140007	CLOSED

TPKEY Information

FTPREF	TPKEY
5.06.C.14.1	PF0000140003

AIRSPEED	1.3VS
ALTITUDE	LOWER AS POSSIBLE
ATA	06
CG	16
DESCRIPTION	CLIMB OEI
FLAPS	0
LG	UP
MANKEY	P003400
POWER	MCT
SELECTED	1
WEIGHT	22000

RUNES Information

R012 R016 R013 R015

AIRCRAFT	OPNUM	RUN
S001	1140	R012

TINI	31475
TFIN	31791
FTPREF	5.06.C.14.1
TPKEY	PF0000140003
MANKEY	P003400
MAN_DESCRIPTION	CLIMB TEST
TP_DESCRIPTION	CLIMB OEI
WEIGHT	22500
CG	18.5

#	AIRCRAFT	FLIGHT	TYPE
2	C295-S001	F111401-AV_BAD	IRF
3	C295-S001	F114001-AV	IRF
4	C295-S001	F111401.0.CDF	CDF
5	C295-S001	F114001.0.CDF	CDF

Figure 11. Info TPKEY

By selecting with a double click on one of the flight files, a plot tool appears containing the time histories of the characteristic flight parameters of the TPKEY manoeuvre.

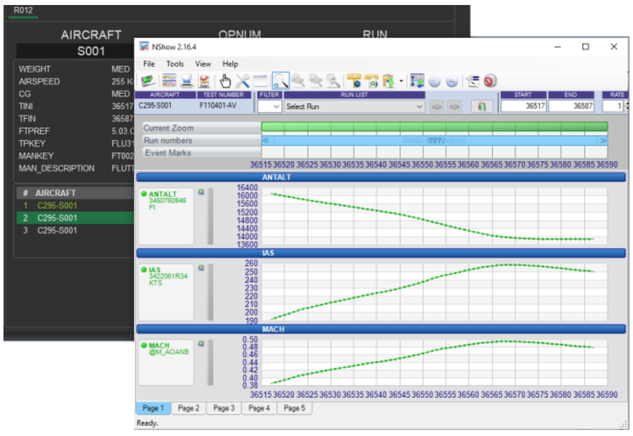
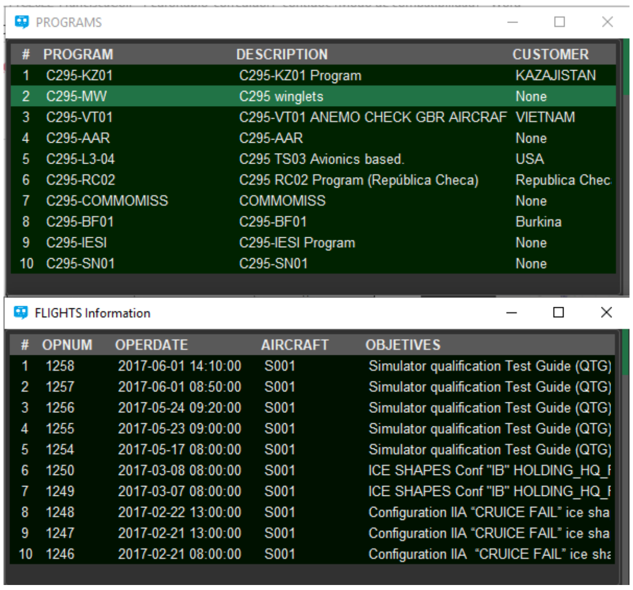


Figure 12.Plot flight parameters

4.3 Program Flights

Other functionality of TABAI is access to the flights information of a specific program.

This information can be obtained from the list of program by pressing the shift key and by clicking on the selected program.



PROGRAMS

#	PROGRAM	DESCRIPTION	CUSTOMER
1	C295-KZ01	C295-KZ01 Program	KAZAJISTAN
2	C295-MW	C295 winglets	None
3	C295-VT01	C295-VT01 ANEMO CHECK GBR AIRCRAF	VIETNAM
4	C295-AAR	C295-AAR	None
5	C295-L3-04	C295 TS03 Avionics based.	USA
6	C295-RC02	C295 RC02 Program (República Checa)	Republica Chec
7	C295-COMMOMISS	COMMOMISS	None
8	C295-BF01	C295-BF01	Burkina
9	C295-IESI	C295-IESI Program	None
10	C295-SN01	C295-SN01	None

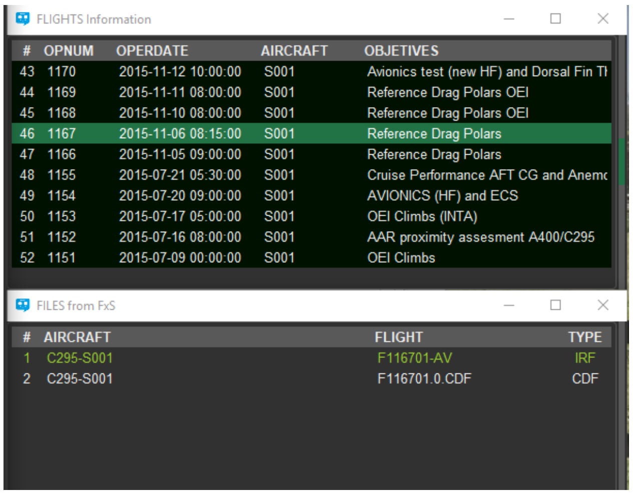
FLIGHTS Information

#	OPNUM	OPERDATE	AIRCRAFT	OBJETIVES
1	1258	2017-06-01 14:10:00	S001	Simulator qualification Test Guide (QTG)
2	1257	2017-06-01 08:50:00	S001	Simulator qualification Test Guide (QTG)
3	1256	2017-05-24 09:20:00	S001	Simulator qualification Test Guide (QTG)
4	1255	2017-05-23 09:00:00	S001	Simulator qualification Test Guide (QTG)
5	1254	2017-05-17 08:00:00	S001	Simulator qualification Test Guide (QTG)
6	1250	2017-03-08 08:00:00	S001	ICE SHAPES Conf "IB" HOLDING_HQ_I
7	1249	2017-03-07 08:00:00	S001	ICE SHAPES Conf "IB" HOLDING_HQ_I
8	1248	2017-02-22 13:00:00	S001	Configuration IIA "CRUICE FAIL" ice sha
9	1247	2017-02-21 13:00:00	S001	Configuration IIA "CRUICE FAIL" ice sha
10	1246	2017-02-21 08:00:00	S001	Configuration IIA "CRUICE FAIL" ice sha

Figure 13. Program flights

This information could have been accessed directly by writing or saying: flights MW.

By double clicking on a flight, the list of flight files appears in a new window.



#	OPNUM	OPERDATE	AIRCRAFT	OBJECTIVES
43	1170	2015-11-12 10:00:00	S001	Avionics test (new HF) and Dorsal Fin Tl
44	1169	2015-11-11 08:00:00	S001	Reference Drag Polars OEI
45	1168	2015-11-10 08:00:00	S001	Reference Drag Polars OEI
46	1167	2015-11-06 08:15:00	S001	Reference Drag Polars
47	1166	2015-11-05 09:00:00	S001	Reference Drag Polars
48	1155	2015-07-21 05:30:00	S001	Cruise Performance AFT CG and Anemc
49	1154	2015-07-20 09:00:00	S001	AVIONICS (HF) and ECS
50	1153	2015-07-17 05:00:00	S001	OEI Climbs (INTA)
51	1152	2015-07-16 08:00:00	S001	AAR proximity assesment A400/C295
52	1151	2015-07-09 00:00:00	S001	OEI Climbs

#	AIRCRAFT	FLIGHT	TYPE
1	C295-S001	F116701-AV	IRF
2	C295-S001	F116701.0.CDF	CDF

Figure 14. Flight files

By selecting with a double click on one of the flight files, a plot tool appears containing the time histories of the generic flight parameters of the complete flight.

5 Conclusion

TABAI is a Text Assistant based on Artificial Intelligence with the objective to be an aid to Test activities.

In the first TABAI version a group of interesting functions have been included in the tool, such as access to information on the status of programs, access to flight information or access to data files including their parameter visualization in a plot tool.

The potential of TABAI is enormous and it is easy to expand its functionalities by following the philosophy of the Chatbot.

TABAI may become a fundamental tool in testing by reducing the time required in those tasks where a machine can do instead using Artificial Intelligence techniques.

6 References

[1] Jere Xu https://towardsdatascience.com/how-to-create-a-chatbot-with-python-deep-learning-in-less-than-an-hour-56a063bdfc44

[2] Parthvi Shah. https://medium.com/mlearning-ai/all-about-rapidfuzz-string-similarity-and-matching-cd26fdc963d8

[3] CMUSphinx Documentation: https://cmusphinx.github.io/wiki/

[4] http://www.speech.cs.cmu.edu/tools/lmtool-new.html

[5] https://at.projects.genivi.org/wiki/display/PROJ/Using+CMUSphinx+and+Training+a+Model+to+Enhance+the+Accuracy+of+Speech+Recognition

[6] https://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/