A WEB SEARCH ENGINE TO FLIGHT TEST DATA

M. Sc. Thesis Proposal

Antônio Magno Lima Espeschit
antonio.espeschit@embraer.com.br
São José dos Campos, SP
February, 04 2008

Abstract

This proposal presents a new methodology to compress, store and retrieve flight test data using artificial intelligence techniques. The concept is exemplified with a mockup database created with digits of number pi simulating continuous and discrete flight test data. These data are compressed using a Bayesian algorithm and dynamic segmentation to define homogeneous segments. Statistical data is calculated for each segment and stored in a relational database to create an inference base for an Expert System. The user interface is similar to an Internet web search engine where the user can type queries and retrieve statistical data from the compressed database and pointers to access the uncompressed data.

Keywords

Artificial Intelligence, Expert System, Database, Data Mining, Flight Test Parameters.

1 Introduction

Embraer is a world leader in the manufacturing of commercial jets with up to 110 seats [3].

With the design of each new family of aircraft, Embraer's Flight Test Division needs to carry out hundreds of tests in Brazil and abroad. In each test, thousands of parameters such as altitude, speed and temperature are measured and analyzed by a team of engineers to meet demanding certification requirements and to develop world-class aircraft.

The testing period can last a little more than one year, with tests conducted several times a day, and in some cases using two or more aircraft simultaneously. By the end of each day, the testing process generates several flight-test files with gigabytes of data each.

Since the storage is limited and the restore process is time consuming, queries over all data is usually impractical when the number of offline flights grows above a threshold.

In this M. Sc. Thesis we will present a methodology that permits maintaining an online compressed view of all flight data that can be searched very quickly.

The web interface is supposed to allow the users to perform queries like "all tests where the parameter P1 is bellow or equal 13 and P2 is between 32 and 40" just typing something like "P1<=13 P2 [32,40]" over the Internet [4].

2 Overview

The Figure 1 shows an overview of the Web Search Engine to Flight Test Data:

- During the test the **Data Acquisition System** (DAS) generates one file per flight test. Each file has engineering units' values of thousands of parameters at different acquisition rates.
- After the flight the **Data Compressor** extracts the original data from files, compresses it and stores the **Compressed Data** in a **Relational Database**. **Uncompressed Data** can also be stored, but it's optional.
- The **End User** has a **Web Search Interface** where he/she can type simple queries like "P1<=13 P2 [32,40]". This interface collects the query and sends it to the **Query Processor.**
- The **Query Processor** interprets and validates the user query, translates it to a valid SQL statement, executes this SQL code and returns the records found (and error messages, if any) to the **Web Search Interface**.
- The **Web Search Interface** formats and presents the data (and error messages, if any) to the **End User**.

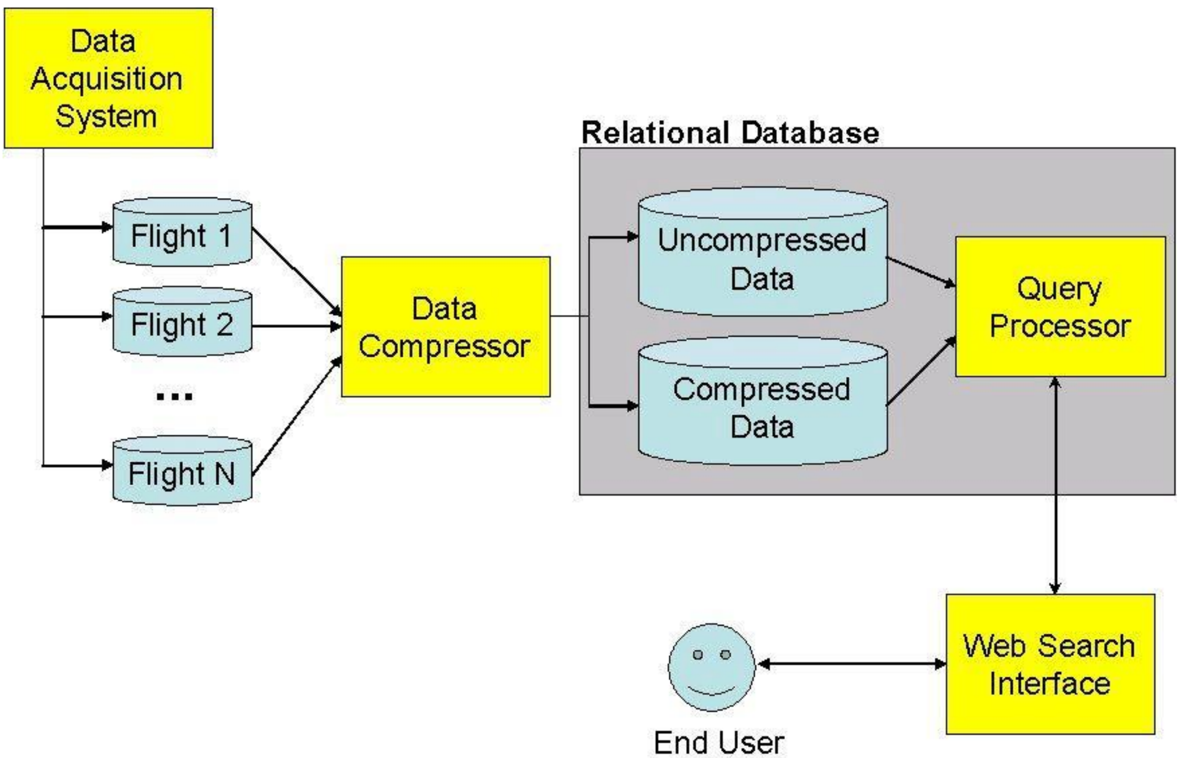

Figure 1 – Overview of the Web Search Engine to Flight Test Data

Instead of using real flight test data, we will simulate it using the digits of number pi [11]. This procedure will preserve the confidentiality of real flight-test data and help other researchers to benchmark this methodology using exactly the same flight-data files that was used to build this system.

Using artificial intelligence techniques [11], the Data Compressor will define “homogeneous segments” [1] using “dynamic segmentation” [7] with “cumulative differences” [6] or “Bayesian algorithm” [5]. Then the system calculates statistical values for each segment and stores it in the Relational Database.

To query the compressed database the user can define “conditions” like “P1<=13” and “P2[32,40]”. The web interfaces validates the syntax of these conditions and send it to the Query Processor.

The Query Processor works in several steps [4]:

- **Translation:** the conditions are translated to valid SQL statements that are sent to the relational database.
- **Execution:** the SQL statements are executed in the relational database.
- **Intersection:** if the user typed more than one condition, the result of each new condition must be intersected with the data from previous conditions.
- **Sorting:** when all conditions have been evaluated, the returned records (if any) are sorted, formatted and returned to the Web Search Interface. If there are many records, the query processor chooses only "the most relevant" data using some heuristics (e.g.: most recent flights first).

In all steps, occasional errors are returned to the Web Search Interface.

3 Workplan

Semester	Year	Activity
1	2007	Approved in CT-234 “Estrutura de Dados, Análise de Algoritmos e Complexidade Estrutural”
1	2008	Approved in CT-200 “Fundamentos de Autômata e Linguagens Formais”
1	2008	Approved in CC-226 “Introdução à Análise de Padrões”
2	2008	Approved in CT-215 “Inteligência Artificial”
1	2009	CE-240 “Projeto de Sistemas de Banco de Dados”
2	2009	English Test
2	2009	CT-205 “Lógica da Incerteza”
1	2010	CE-276 “Programação para Internet”
1	2010	CT-300 “Seminário de Tese”
2	2010	CT-300 “Seminário de Tese”
2	2010	Thesis defense

4 Expected Results

A paper has to be presented in the European Telemetry Conference in October 2009.

A beta version of the system has to be deployed in Embraer until November 2010.

References

[1] AASHTO. *AASHTO Guide for Design of Pavement Structures.* Technical Report, American Association of State Highway and Transportation Offices, Washington, DC. Apendix J. 1986.

[2] CADKIN, Jennifer. *Understanding dynamic segmentation.* ArcUser Magazine. October-December 2002.

[3] SYBASE Press Release. *Embraer Success Case.* http://www.sybase.com/detail?id=1052732. Visited: Sept. 7, 2008.

[4] ESPESCHIT, Antônio M. L.; PENNA, Sérgio D. *A web search engine to flight test data.* In: European Telemetry Conference. Munique. 2008.

[5] FRIDTJOF, Thomas. *Automated road segmentation using a bayesian algorithm.* Journal of Transportation Engineering 131(8), pp. 591-598. 2007.

[6] FRIDTJOF, Thomas. *Generating homogeneous road sections based on surface measurements: available methods.* Proc. 2nd European Pavement and Asset Management Conference, 21-23 march 2004, Berlin, Germany. Paper 48. Available at http://www.vti.se/EPiBrowser/Publikationer/S360A.pdf. Visited: Sept. 7, 2008.

[7] FRIDTJOF, Thomas. *Statistical Approach to Road Segmentation.* Journal of Transportation Engineering 129(3), pp. 300-308. 2003.

[8] PENNA, Sérgio D.; ESPESCHIT, Antônio M. L. *An end-to end approach to test data definition, acquisition and distribution.* In: European Telemetry Conference. Garmisch-Partenkirchen. 2006.

[9] PENNA, Sérgio D.; ESPESCHIT, Antônio M. L. *Flight test data handling through extensive use of parameter groups.* In: European Telemetry Conference. Garmisch-Partenkirchen. European Telemetry Conference. 2004.

[10] PENNA, Sérgio D.; ESPESCHIT, Antônio M. L. *A web-oriented approach to data distribution and work collaboration in a flight test environment.* In: European Telemetry Conference, Garmisch-Partenkirchen. 2002.

[11] RUSSELL S.; NORVIG, P. *Artificial Intelligence - A Modern Approach.* Prentice-Hall 2nd Edition. 2003.

[12] *Super_Pi for Windows* (calculation of pi up to 33.55 million digits). Available at http://www.geocities.com/hjsmithh/Pi/Super_Pi.html. Visited: Sept. 7, 2008.