

# **ENGINEERING THE INFORMATION FOR A WEB- BASED TRAINING SYSTEM FOR THE USE OF AIRBORNE VEHICLES.**

## **PROGRAMMING PROJECT PROPOSAL.**

**UNIVERSITY OF OSNABRÜCK / UNIVERSITY OF TWENTE.**

Presented by

Jorge A. Martínez Malacara  
Student of the Master in Information Engineering  
Matriculation number 917654

Mecklenburger Str. 2 Zi. 221  
49088 Osnabrück  
[jmartine@uni-osnabrueck.de](mailto:jmartine@uni-osnabrueck.de)

Osnabrück, Germany  
March 15<sup>th</sup>, 2004.

## ***Introduction.***

Many times large amounts of information need to be processed and analyzed in order to analyze patterns and compare them with expected patterns. A very good example of it could be the black boxes used by airplanes.

Data generated by multiple sources and sensors is recorded constantly and it can be retrieved and analyzed thereafter. This analysis occurs generally after a flight incident (or a crash). Its primary purpose is to provide with information which could help to find the causes which lead to the situation in which the airplane was involved.

The concept of a black box can have many other applications, especially as an extension to live monitoring of operations. The main advantage is that the information generated by a device is stored for a certain amount of time, so it can be retrieved and analyzed.

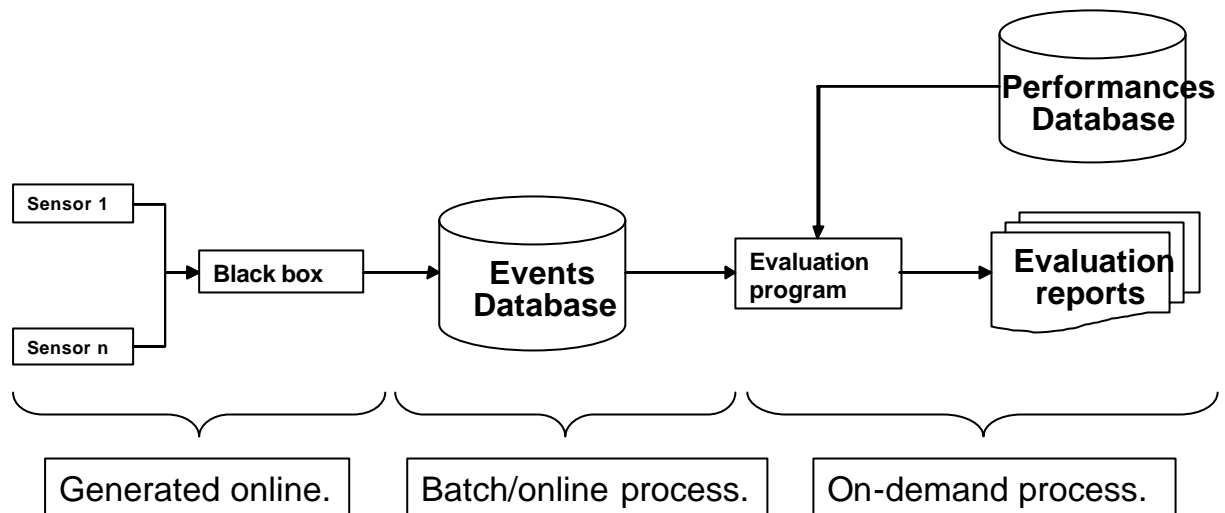
Commonly one thinks of a black box as the source of data which can lead to know the causes of an incident or an operation malfunction. However, these systems can be very useful testing and simulating systems, or training and evaluating personnel. The latter application becomes of interest in this Programming Project, due to the needs of engineering of information under certain operating conditions.

## ***The problem.***

The black box paradigm is all about storing data obtained over a certain period of time from an array of sensors. After a certain period of time, the oldest data is overwritten by new data. However, the information itself is never analyzed until a malfunction in the monitored system occurs. A black box turns to be only a very useful warehouse of clues, which is used mainly to determine the causes of a problem.

As we stated above, the information can be used personnel training, however, the task requires very careful information engineering. The basic problem demands to translate the information from the sensors (or the black box) into a format which could be interpreted and read. We also need to transmit the data to a data warehouse where we have the ability to analyze the information. The analysis task implies the pre-existence of a knowledge base or a set of sample data which could be used as a reference for comparison. The creation and/or use of that knowledge base make the task even more challenging.

The architecture of such a system would look as follows.



As we can see, there are 3 phases in the information flow. The first stage is generated online, with the sensors feeding the black box. The second process is batch or online, depending on whether there is a permanent communications link between the black box and the events database. The third stage is the evaluation process itself, when the information contained in the events database is compared with the expected/allowed performances. It could be possible to update the performances database afterwards, given that the nature of the monitored system allows it.

Let's define log as the set of data generated by the black box and transmitted to the events database.

Today's technologies allow concentrating the analysis task in one place, but it is required to transmit the log, which can turn into an information engineering problem.

## ***The Information Engineering Problem***

There are several challenges from the point of view of Information Engineering. On one hand, we can find ourselves creating huge logs of information, which require be compressing and transmitting over the internet. On the other hand, the logs require detailed information but at the same time they have to be as small as possible. The information then has to be analyzed and interpreted by a program, which has to compare such information to the parameters given by the performances database.

## ***The proposed system***

It is not an easy task to find a system with sensors which can be analyzed in such a way that in 6 weeks a working log analysis system could be implemented. In order to focus in the information engineering tasks, an implemented system had to be found. A very good alternative is the Microsoft's Flight Simulator (MSFS). This software simulates a real airplane's operation to a very high level of detail. In addition to this, there are hundreds of programs available which increase the functionality of the program and enable to capture data from the game.

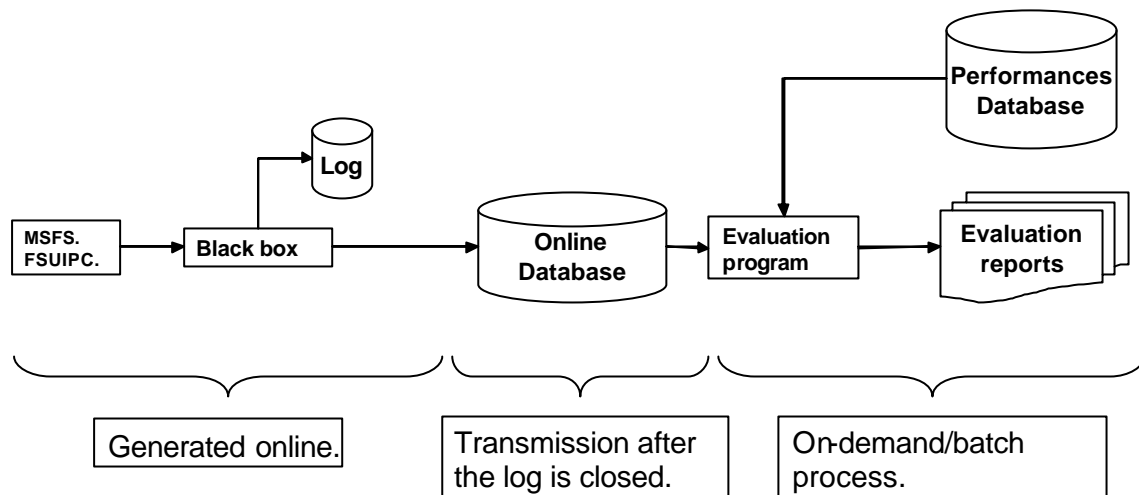
Therefore, a black box system can be implemented within a short period of time, given that there enough documentation available for such a task. Such a system should be programmed using C/C++ or Visual C++. Other programming languages can also be used, but due to the lack of time, a powerful programming language had to be chosen.

On the other hand, there has to be a program or a script which allows decoding the log generated by the black box system and compares it with given parameters from the performance's database. I chose PHP to achieve this goal.

After a small research trying to find the ways to read the information from the Flight Simulator, 2 programs turned out to be useful. The first one is called "netpipes" and it is provided by Microsoft. The program contains header files in C++ language which can be used to read the information from within the Flight Simulator in the form of offsets. However, a serious limitation was also discovered. Microsoft has changed the internal offsets of the Flight Simulator which can be read from the program. Therefore, there exists a version of netpipes for each of the versions of the Flight Simulator since FS98. In addition to this, it was discovered that a significant amount of Flight Simulator's users have old versions of the program (FS98, FS2000, FS2002). This would imply almost a different black box program for each one version of the MSFS.

There is another utility, currently very popular among developers for the MSFS. It is an internal Dynamic Link Library (dll) created by Pete Dowson, called FSUIPC (Flight Simulator Universal Inter Process Communication). This DLL maps the offsets from each one of the last 4 versions of Flight Simulator into a single offset table, which can be read by external programs. There exists documentation available for several programming languages on how to implement applications using this library; therefore it was selected as the best solution to the information gathering problem.

The system will therefore look like the following diagram:



**Black box program** a C/C++ program which reads information from the MSFS via FSUIPC. The information is transferred to a log file. When the flight is finished, the log is evaluated and its size reduced. Then the software sends it to the internet to a predetermined online database.

**Evaluation program** It's a series of PHP scripts. They evaluate the received logs and compare them to a performance database. According to the differences found, an evaluation report is generated for every single log.

This is the proposal I have for the programming project. Given its acceptance, it will be developed within the timeframe of 6 weeks. There will be weekly reports where all the incidences during the developing process will be detailed. In addition to this, and given that there is enough time, there will be a series of alpha tests of the black box program made by Flight Simulator users. Their logs will be kept as a testing source for the PHP scripts.