

TODO: insert a title here

Author:

Andrea G.B. DAMIOLI

Supervisor:

Dr. Germano BONOMI



Master Thesis

Abstract

Master degree in Computer Engineering

TODO: insert a title here

by Andrea G.B. DAMIOLI

The AEgIS Experiment at the CERN aims to verify the weak interaction principle for antimatter. This document talks about "gAnWeb", a web application designed to simplify the analysis of physical data under the AEgIS experiment. This analysis can be performed using Root Data Analysis Framework by the Linux Terminal using an application named "gAn", but a graphical interface can ensure a better user experience, ease the user training and improve the productivity. This document analyses from the Human Machine Interaction's point of view the production of this graphical interface.

Contents

Abstract	iii
1 Introduction	1
1.1 User friendly Data analysis: gAn Web	1
2 Requirements	3
2.1 Functional requirements	3
2.1.1 Early version	3
2.1.2 Late version	5
2.1.3 Ambiguities (and related solutions)	7
2.2 Unfunctional requirements	7

Chapter 1

Introduction

1.1 User friendly Data analysis: gAn Web

gAn is a program that aims to analyse huge amount of data related to the AEGIS experiment at the CERN.

This program receive in input a folder containing several terabytes of files in ".root" format, and parameter named "run parameter", that identifies the information in which the users are interested. A file ".root" is a file produced by a variegated group of sensors in a complex machine that accelerates particles and lets them crash together. This sensors produce .root files continuously (8 hours per day). The time in this experiment is divided in "runs" (a run lasts about 140 seconds), so the user, by the run parameter can tell to gAn in which time slice he is interested.

The .root files can be analysed using a framework named ROOT Framework, that consists in a lot of libraries specialized in high-energy physics analysis, and an interpreter able to understand a C++ script. Actually, gAn is the sum of the Root Framework plus a lot of C++ scripts. The goal of gAn is reduce the huge amount of raw data in input in a little amount of scientifically interesting data in output. To do this it has to filter data, understand which of them are scientifically interesting, chose the parts that are related to the run selected by the user (by the run parameter), elaborate and compare them, and make advanced statistical analysis on them. gAn can be called using a common linux terminal, using a command with parameters.

The output of gAn consists of a single text file with computed, organized data, and a folder of images in png format. This structure (root files in input, data analysis using Root, images, organized and selected data in output) is very common in the CERN's experiments.

The output of gAn is quite understandable by an experienced physicist, but it is disorganized, complex for an untrained user, and the terminal interface can be surely improved using some more user friendly technologies.

GAn Web is a web application, that aims to create a user friendly web interface, based on the most important human-machine interaction principles, between the users and gAn. A web interface can improve it in two ways:

1. gAn is a stand-alone program based on Root, installable on the user's machine; the user has to install the correct version of Root to avoid compatibility problems (Root is still not perfectly version independent: different versions can lead to different behaviours). Furthermore, this kind of program is continuously changing, the performed analysis is continuously improved, so the installed version of gAn is not final and unchangeable, and the user must often update it. Instead, a centralized version installed on a server, with services accessible from a normal browser by the user can avoid (at least reduce) this kind of problems and be more usable.
2. a Linux terminal interface is practical for expert users, but a web based interface can be more attractive for new users, and, if well done, can be easier to use. It is important to notice that the users are physicists, not necessarily specialized in computer science, so, create a friendly and easily learnable interface can avoid them problems and time wasting.

The goal of gAn Web is to allow users to do analysis through a more friendly web interface, without install nothing on their machine. In the following image there is a schema that shows how this program is organized.

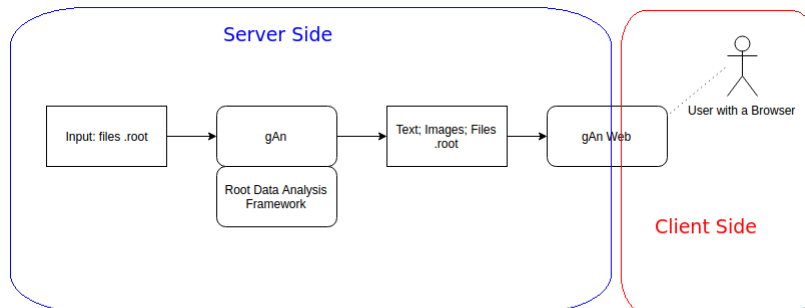


FIGURE 1.1: gAn - gAn Web simple scheme

Chapter 2

Requirements

First of all is important to analyse which are the requirements of gAn web (and how gAn web can improve the performances of gAn).

2.1 Functional requirements

The definition of functional requirements aims to specifying in detail what the web application can do, and in which way an user can use it. The development of gAn web is divided in two stages: an early stage, more simple, with basic functionalities, just to investigate what are the best ways to implement the functionalities and to test with a little group (2-3) of users if this software can really be useful and which functionalities are really important; A late stage, more complex, with some complex functionalities obtained listening the request of the first group of users; It is important to notice that the second stage (the late one) is a never-finished stage: the needs of the users are permanently changing and evolving, the application is designed to be adaptable, and to try to satisfy the unknown needs of future users. Following are exposed the requirements of the two versions:

2.1.1 Early version

The first version (from now "gAn web v1") is a very simple application: instead of access the program by a linux terminal like gAn,

in gAn web v1, the user can use the program through a graphical interface. The requirements of this version are the following:

1. The user, in the homepage, can chose the run (only one run, for the moment) in which he is interested, using a input field. This field has validator, able to understand if the run number is inserted, if it is effectively a number, and if it is in an acceptable range. The user receives an explaining and precise error message directly on the homepage if the input field is empty and if the inserted value is not acceptable.
2. The user can start the program with a single click, by a button (usable only if the inserted number is valid).
3. When the program is executed the user can see the text output on the screen. This text is clear for a physicist (it is not clear for a person who doesn't have a specific preparation).
4. When the program is executed the user can see the output images by clicking a button that link to a images-page. The images are ordered and organized by groups (the groups are related about which sensor takes the information necessary to create the image). The user can decide if he prefers to see the image in a little, medium or big format. The user can also decide if the images are distributed in the screen vertically or through a "carousel layout". The user can access the image in full-screen by clicking on it: he is redirected to a page with the image shown in full screen, and can return back to the all-images page by a return button.
5. The user can modify a configuration file (a .txt file on the server), by a web interface. In this files there are some values the need to be setted (otherwise it use default values), and the user can do it by radio buttons (in this way he is forced to chose valid values). This configuration file can modify the way in which gAn works and modify the resulting output (both the text and the images).

2.1.2 Late version

The late version is more complex. It was born from the tips and the observation of the first users. The modifications are not numerous, but there are a lot of additions of new features. All the new required characteristics are exposed following:

1. The user can insert multiple runs: separated by a semicolon (but in case of errors the system can automatically correct them replacing symbols like "-" or ";" or "." with semicolons and giving a more robust service). These runs can be inserted by an input field or by a range select button: this button opens a "modal" that allows the user to choose the first run and the last, and automatically insert the comprised runs (for example, if the user inserts 30000 and 30010 the system inserts automatically all the run numbers between 30000 and 30010). This modal has a validation system, that ensures the correctness of the inserted values.
2. The user can choose which version of gAn to use for the execution of the program. In this moment exist 2 versions, but in the future they can become more. They are externally very similar, the differences are the algorithms in the program, but they give a different output (different output but in the same format: text and images). Another difference is that the configuration file is not written in txt, like in the early version, but it is written in xml (see next point).
3. The configuration file is not only in text format, but also in xml format (it depends on the selected version of gAn). This fact ensures a stronger structure, and must be transparent to the user (he mustn't see differences).
4. The user can choose what version of Root he wants to use for the program. Theoretically different versions of Root are perfectly compatible, but in practice each version of gAn is designed to work with a particular version of Root and to avoid problems it was specifically requested to the gAn web designer to allow the user to choose freely which version of Root to use among the installed versions on the server.
5. The user can save images on his hard disk: he can choose from the shown images in the images page an image to download

by clicking on a specific download button near the image. Furthermore there is another button "Download All" with whom the user can simple download all the output images.

6. The user can download a reduced version of the root file with informations about the images and the results: gAn produce this kind of files as "half-processed" during the computing, and it is not a problem to save this on the hard disk of the server in a specified folder. For an expert user can be scientifically interesting have this file (this root file contain more information than the output, the most of this information is useless (it is an "half-processed" file) , but sometimes an expert user can find something interesting), so the user must have the opportunity to download this.
7. The first little group of user prefers the dropdown menu to the radio button, so all the radio buttons in the program are replaced by dropdown menus.
8. The user has to access not only to a png image, but to a root-image. This kind of image is interactive: the user can with a left click of the mouse (a continued click, like the "dragging") select parts of the image and zoom them, and with a right click do dynamically some kind of image processing (set colors, chose what kind of chart to show, modify the chart legend, translate in a 3D space the image etcetera). All of this must be done by the user through a browser window. This requisite seems to be very complex, but Root provides libraries (these libraries work well but they are poorly documented) to interact with Javascript, and can in some way resolve the problem.
9. In the homepage the user can see the run number of the last root file produced by the machine, and its creation date and time (so, he can understand what is the maximum of the range of the insertable numbers)

In the late version there was another functional requisite: ideally the user should have been able to select a gAn version also if not installed in the server machine: in this case the system should have been capable to automatically search on the AEgIS Gitlab repository the correct version (if existing), download it, unpack it in the server, and use it to execute the program. After some discussion this requirement has been cancelled, because it was considered complex, basically useless, and potentially harmful (on the branches of the repository there are untested

and incomplete versions, that can create if executed wrong outputs, so wrong scientific results). At this moment installing manually the stable versions of gAn on the server seems to be a more smart way to work.

2.1.3 Ambiguities (and related solutions)

At least a point seems to be quite ambiguous:

The textual output of gAn needs to be formatted in some way to be more organized and clear? The answer is difficult: for a non-physicists this output seems to be disordered, too long, with too many groups of informations, and very difficult to understand, but on this question all the physicists questioned answered that the output is perfectly clear and doesn't need to be modified or improved in any way. The only requests of the users were about the font and the font-size. Anyway, in the second version, in case of multiple run selection, there is a "navbar" that allows the user to show only a run-result per time.

2.2 Unfunctional requirements

unfunctional are... definition

lists: no crash (at least don't kill apache), no install