JAVA Graphical User Interface for Simulations in the ATLAS/Tilecal experiment

H.Wolters [a,b], F.Colaço [b], A.Amorim [c]

The ATLAS/Tilecal collaboration

[a] *Escola Superior de Ciências e Tecnologia, Universidade Católica Portuguesa, P-3080 Figueira da Foz, Portugal*

[b] *LIP, Universidade de Coimbra, P-3000 Coimbra, Portugal*

[c] *LIP and CFN, Universidade de Lisboa, P-1699 Lisboa Codex, Portugal*

An object oriented graphical user interface is developed to control the parameters of an existing complex simulation package. The choice of JAVA as the basic programming tool allows running the interface on almost any client in the world without specific installation. Thus, it can be seen as a general example of providing up-to-date user interfaces to historically grown, complex software systems.

Key words: GUI's; WWW applications; Java.

1 Introduction

Simulation calculations in the ATLAS/Tilecal experiment[1] are based on a very complex, historically grown Monte Carlo simulation software[2]. The user specifies the simulation parameters by means of an ASCII file. The lines of the file are commonly called "datacards"; each one contains one instruction, consisting of a four letter keyword and a list of numerical parameters[3]. A typical example of a datacard that specifies part of the beam kinetics of the simulation would look like

KINE 2=100. 3=8. 6=0. 7=0.011 8=10. 9=0.011

This specifies a 100 GeV pion beam with incident angles $\phi = 0^o$ and $\theta = 10^o$ with dispersions of 0.011^o for both. In order to edit this card, the user would look up the user's manual and find that the beam energy is specified by the second parameter in the KINE datacard, in GeV, and so on. The typical procedure is editing an existing datacard file that is known to contain the basic setup parameters needed for the simulation, and then adapting the parts of the setup that have to be changed.

This procedure has a lot of inconveniences. First, for a newcomer it is almost impossible to analyze the content of a current setup, because of the big amount of cryptical information, each of which needs a careful lookup of a reference manual. Second, it is very easy to accidently delete one datacard during the editing, thus ending up with an unexpected specification where the corresponding part of the current setup might be replaced by some default values not adequate for the actual simulation.

It was thus desirable to supply a new user interface that allows the user to specify the simulation parameters in a more obvious way. We decided to develop a user interface that simply wraps around the well established concept of data cards, i.e. essentially supplies an object oriented editor for the datacard file. This concept has the advantage that we do not need to modify directly the code of the rather complicated simulation software.

2 General Concept

The application *Susi* (**S**imulation **U**ser **I**nterface) allows the user to specify all simulation options in a transparent way, such as detector geometries, signal read-out options, beam kinetics etc. *Susi*, then, translates the specified options into the standard data card format of the simulation software.

In view of the large number of users of the simulation software, spread all over the world, with all levels of computer expertise, we have decided to opt for an application that can be run in a WWW browser. There are three basic advantages from this choice. On one side, we avoid the problem of maintaining *Susi* for the many different platforms used inside the HEP collaborations. On the other side, we do not have to annoy users with the necessity of having to install a software package on their home system, which is specially relevant for the less experienced users that are one main target of the new interface. This, in addition, simplifies a lot the management of the code, and the users do not have to worry about upgrading to new versions. The application needs only to be installed on one central server, and, nevertheless, can be run from any machine that is linked to the WWW and has a suitable browser installed.

Various attempts have been studied as of which is the best choice of development tools for the interface, namely CGI, JAVA-Script and JAVA. The actual application is running in JAVA. It can be started by any WWW browser that supports JAVA, as for example Netscape.

Apart from the advantages mentioned above, the WWW browser environment also reveals some quite basic problems. Due to security reasons, JAVA does not permit to write to or read from the client disk if running in a browser environment. Even I/O with the server disk is not straight forward. The point is that, in an open network environment, neither the server can trust the remote user, nor the client host can trust the server software, and there is still no well-defined authentication concept existing. However, JAVA is still a very young tool, and each new upgrade has revealed considerable improvements. For the time being, we use the clipboard of the local window system to handle the necessary user I/O.

3 Implementation

The simulation parameters are logically divided into eight menus. Each menu covers one conceptional area, like the beam kinetics or the detector geometry. Figure 1 shows the startup window. It is obtained by simply opening the URL `<http://piranha.fis.uc.pt/susi>` in a Java-compatible browser, e.g. Netscape, on any computer that is linked to the Internet.

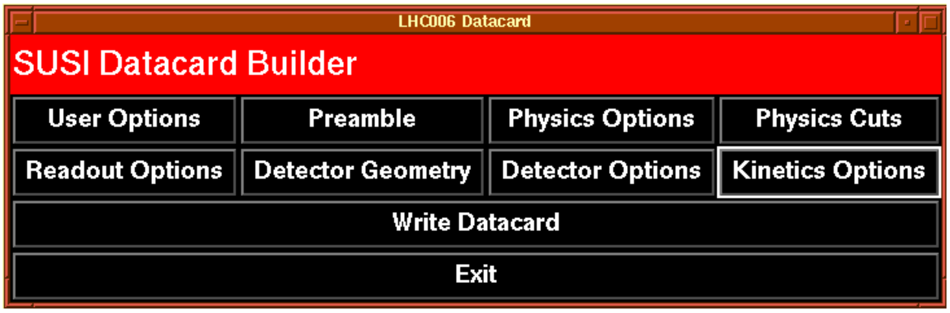

Fig. 1. *Susi* Main window

At the moment, eight sections exist, which cover the datacards presently used in the ATLAS/Tilecal simulations. Figure 2 shows part of the menu that is used to specify the beam kinetics.

After having specified all simulation options, selecting "*Show Datacard*" will pop up a window containing the corresponding datacard file for the simulation program. From there, an option entitled "*Select*

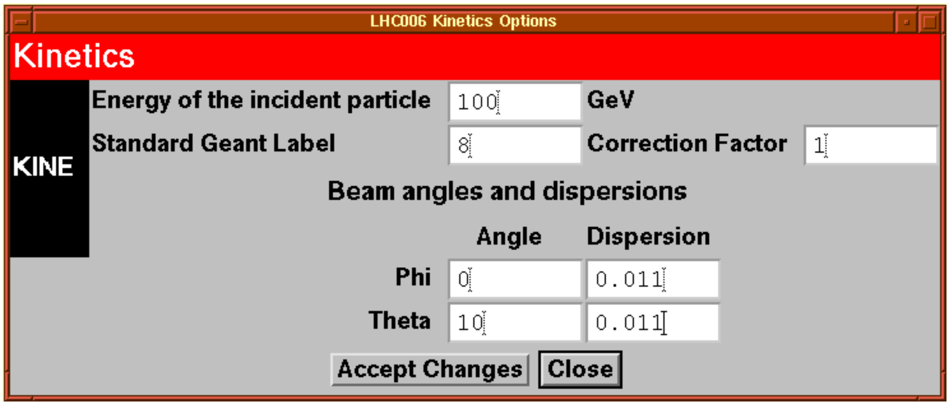

Fig. 2. *Susi* — Part of the Beam Kinetics menu

all' copies the datacard to the clipboard, from where it can be pasted into a text processor on the local computer and finally saved to the local disk. This is a workaround due to the security restrictions for Java applets mentioned above. If *Susi* is invoked locally, i.e. not via a WWW server but running as a normal JAVA executable on the local machine, the datacard can be loaded and saved directly via a file dialog. This local version also contains a datacard file parser, that reads an existing datacard and allows modifying it.

4 Outlook

For the time being, *Susi* offers an object oriented approach to the creation of the datacards, still with workarounds though, especially concerning file I/O. Due to the modular structure it is very easy to extend the program to cover other groups of datacards. The next planned steps of improvements are

- building a datacards database, both at a local (client) and at a central (server) level. For this purpose, the problems of server and client I/O have to be resolved;
- presentation of a geometrical view of the beam kinematics, and positioning of the detector relative to the beam with the mouse;
- calling the simulation program directly from *Susi*, either on the local computer or on a server;
- bookmark links to the online documentation of the simulation package for each parameter.

References

[1] ATLAS/Tilecal Collaboration, *ATLAS Tilecal Design Report,* CERN/LHCC 96-42 — ISBN 92-9083-091-3
http://atlasinfo.cern.ch/Atlas/SUB_DETECTORS/TILE/tilecal.html

[2] ATLAS/Tilecal Collaboration (F. Ariztizabal et al.), Construction and Performance of an Iron–Scintillator Hadron Calorimeter with Longitudinal Tile Configuration, *Nucl. Instr. Meth.* **A 349** (1994) 384

[3] GEANT User Manual and FFREAD package therein, CERN Library,
http://wwwinfo.cern.ch/asd/geant/index.html