DRDC/94-28

Letter of intent to the DRDC

May 26, 1994

H.Fesefeldt
III. Physikalisches Institut, RWTH Aachen, Germany

S.Banerjee
Tata Institute of Fundamental Research, Bombay, India

M.Ballintijn, J.P.Baud, R.Brun, O.Couet, S.Giani, F.Hemmer
A.Nathaniel, A.Osborne, S.Ravndal, F.Rademakers, L.Robertson,
J.Shiers
CERN CN division

O.Schaile
Freiburg University, Germany

M.Benard
Hewlett-Packard, Geneva

K.Amako, J.Kanzaki, T.Sasaki, Y.Takaiwa
KEK, Tsukuba 305, Japan

L.Urban
KFKI, Budapest, Hungary

Y.Nakagawa, T.Yamagata
International Christian University, Mitaka 181, Japan

M.Maire
LAPP, Annecy, France

P.Gee, B.Greiman, D.Hall, C.Tull
Lawrence Berkeley Laboratory, USA

Introduction

The Application Software and Data Base group in CN Division has been involved for many years in the development and support for many packages and tools widely used in the HEP community. More recently, a paper[1] proposing an evolution of the CERNLIB environment has been discussed at COLLECT and presented at the LHC software day at CHEP94. The CN/ASD group and collaborators are proposing to actively collaborate to the design and development of a seamless environment for the simulation, reconstruction and analysis stages of the LHC experiments in the framework of an Object Oriented technology. Some of the ingredients of this framework presented in Figure 1 could be:

- A simulation kernel based on the extensions of the GEANT system.
- A common detector and event visualisation system.
- A common event data model.
- A common I/O system based on ZOO. The framework must support transparently the Distributed Computing model. Each module of the framework can run anywhere in the world using ZOO (via CORBA) to transfer the objects. A consistent data storage and data management model must be designed.
- A common User Interface system based on KUIP and its planned extensions.
- A common Data Analysis environment based on the evolution of the PAW and PIAF systems.

It is our intention to submit three proposals to the DRDC in the coming weeks. These proposals are not specific to one particular experiment and are aimed to be useful to both the existing experiments as well as the proposed LHC experiments.

The list of collaborators is not yet complete - we hope to attract more manpower especially for the ZOO project. In addition it would be natural to invite the PASS project collaborators to join in the Very Large Database project.

Following is a short description of each proposal together with the list of persons and groups who have expressed interest in joining this work. Finally we indicate the relationship between these projects and support for current software packages, and P55.

Proposed Software Framework

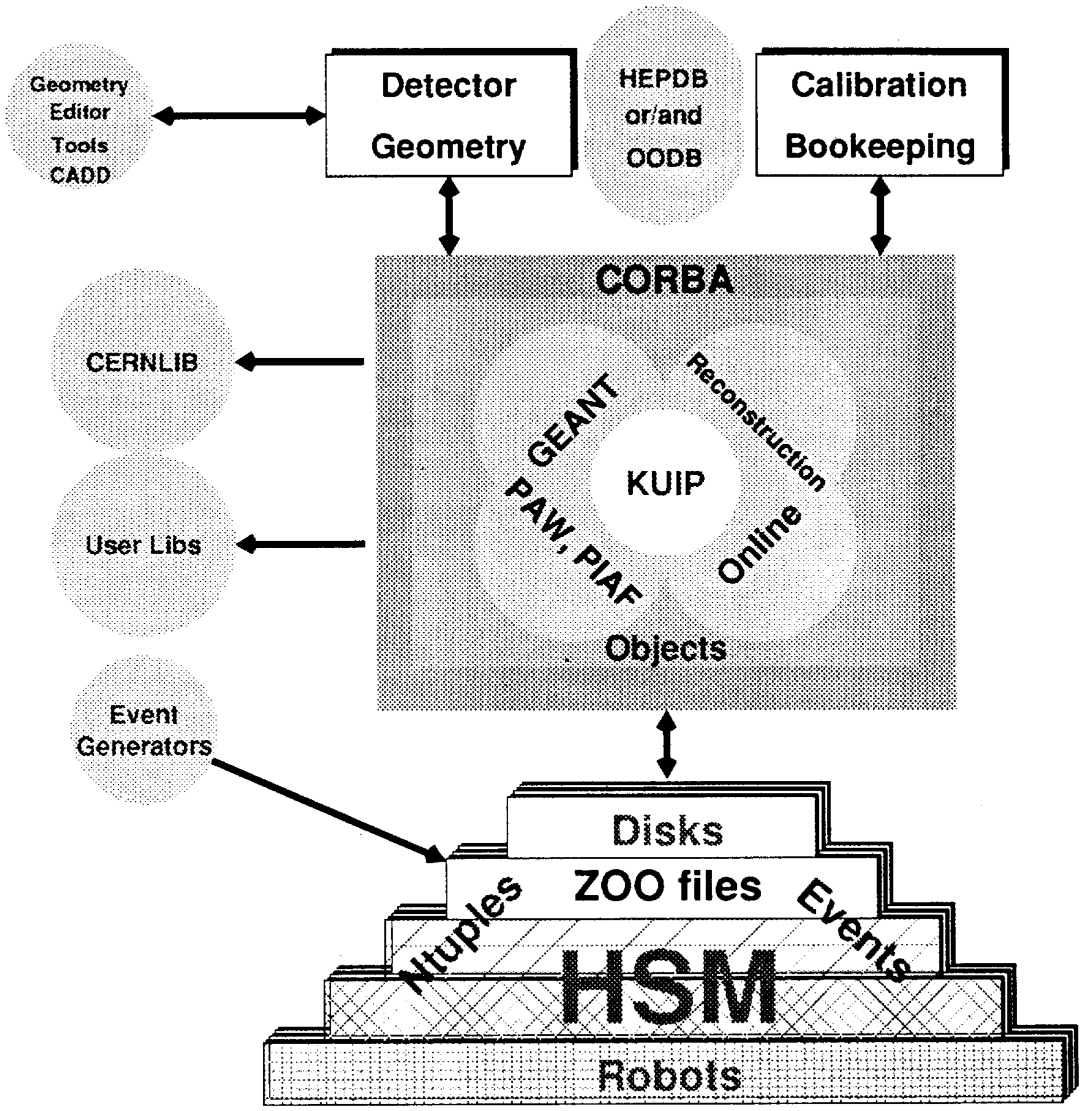

Figure 1: The proposed framework

1 GEANT

We are now planning important developments in GEANT, taking into account the evolution of the environment and the impact of the new languages and the data models. Algorithmic and Structural changes are required.

Proposed structural changes

- Redesign the **data model** currently based on ZEBRA data structures in an Object Oriented approach. A prototyping exercise[2] to define the classes and their relations has already been done in collaboration with the PRODIG project at KEK. The consolidation of this work must take place in close collaboration with the LHC groups. The new data model must be seen in the global frame of simulation and reconstruction.

- Implement a comprehensive detector and event viewing system based on an upgrade of the existing interactive version. This requires a very close cooperation between the experimental groups and the application programmers currently developing the KUIP and HIGZ packages.

- Redesign the interface to the **User routines** called at tracking time. It should be possible to call Fortran (77 and 90) routines or C/C++ procedures from the GEANT kernel. We believe that it is possible to preserve backward compatibility with the existing version of GEANT for those intending to continue for a few years with their existing Fortran-based interface. This approach has also the advantage that it can test the new code by checking it on existing detectors.

- Re-engineer several parts of GEANT to facilitate the maintenance work and cooperation with new potential collaborators.

- Rewrite the GEANT kernel, geometry and tracking, in C++, taking advantage of the existing logic.

- Rewrite the I/O package as soon as the ZOO I/O system becomes available.

Proposed algorithmic changes

- Upgrade the geometry package to improve speed, reliability, simplicity and power.

- This upgrade should facilitate the interface with the CAD systems. Interesting results have already been obtained by the CADD project. However, an automatic and symmetric interface between GEANT and CAD systems still requires a lot of work.

- Rewrite an important part of the tracking system. It is necessary to remove the fraction of the code corresponding to some old tracking algorithms never used by the experiments because the gain in time has never been substantial. This clean-up should in itself improve performance and make for a considerable simplification, thus easing the maintenance. Tracking precision will be improved by using almost exclusively 64 bit arithmetic throughout.

- Consolidate the physics processes, in particular the hadronic physics package. This implies a better evaluation of the range of validity of the physics in GEANT by comparing with experimental data. These processes will need to be adapted to the important structural changes foreseen in the GEANT kernel.

- Some attempts have been made in the past to include a global parameterisation scheme in standard GEANT. These prototypes have, however, never reached a stage of sufficient maturity to allow them to be included in the official release. This work should be resumed in collaboration with the new experiments.

Using GEANT as an education and training tool

The GEANT program is a powerful tool to teach the experimental aspects of High Energy Physics at Universities. The visualisation of the detectors and the details of the physics processes at tracking time could be an attractive and instructive complement to abstract lectures. The continuous simplification of the interactive user interface is progressively making GEANT a potential educational tool.

In the context of the large LHC collaborations where we expect a continuous flux of new physicists, it is also extremely important to develop adequate tools to facilitate integration and a fast understanding of all aspects of a detector. With the impressive progress now taking place in the areas of animation and **virtual reality**, it becomes possible to develop a **real-time navigation system for a detector based on** GEANT. A first prototype illustrating the possibilities of this technique has been demonstrated at CHEP94. This tool could also be used by the collaborations to better illustrate and explain their detector in their public presentations. It could also facilitate communication with non-physicists.

The GEANT project will be a collaboration between:

- The CN/ASD GEANT team: S.Giani, S.Ravndal, R.Brun

- The members of the PRODIG project at KEK and Tokyo: K.Amako, J.Kanzaki, T.Sasaki, Y.Takaiwa, Y.Nakagawa, T.Yamagata.

- H.Fesefeldt Aachen

- L.Urban KFKI Budapest

- M.Maire LAPP

- Physicists in ATLAS and CMS responsible for detector simulation.

- the VENUS project for **virtual reality** at CERN: S. De Gennaro

2 Analysis of very large data bases

The long term aim of this project is to provide a standard environment for the storage, access and analysis of the Petabyte data bases announced for the LHC era. We see that as an iterative and pragmatic process in strong collaboration with the existing LEP experiments and the coming fix target experiments.

For many years the HEP data model has been based on a cascade of sequential files. From raw data down to DSTs, the analysis programs were designed to loop **sequentially** on all these files, possibly with a fast skipping mechanism to read only the interesting events. At each of these stages histogram files could be produced. Currently micro-DSTs have been conveniently replaced by row-wise Ntuples and many collaborations are in the process of converting their mini-DSTs to column-wise Ntuples. The Ntuple concept greatly facilitated the data analysis phase by offering in an interactive system like PAW many of the advantages of relational data base management systems. With Ntuples, direct access to any event in the file considerably improves the search mechanism. In the case of column-wise Ntuples, only the data used in the query is read from the file, thus saving a non negligible amount of I/O resources.

The conventional flat-access model for one event must be reconsidered in the light of an Object Oriented approach in order to optimize the access to the various components of an event. Column-wise Ntuples should support objects without the performance penalty of the existing object-oriented data base management systems. There are several ideas on how to combine the advantages of a relational model and the OO paradigms. Many data base providers try to solve this problem. However, it is unlikely that they can propose a system flexible and efficient enough to handle the gigantic data bases that we expect for LHC. We have in our hands a lot of experience in dealing with very large data bases. This development must take into account parallel file systems, hierarchical file management and migration and obviously parallel architectures.

The Online and Offline architectures are clearly affected by this data model. The data acquisition systems will not write to tapes, but instead will accumulate data on disk-resident hierarchical data bases. It will be the task of the Hierarchical Storage Manager (HSM) to migrate these files to a sequential medium. The HSM includes several levels of **cache** transparent to the application. It is essential that any future storage management software implement the concept of dynamic hierarchies and provide a transparent migration of data from one type of medium to another.

The data base model is compliant with the emerging standard CORBA. It supports concurrent requests for read or write via the Object Request Broker. It is also the task of ORB to optimize resource allocation.

The successful developments that have taken place in the context of the emerging PIAF architecture and the theoretical work done in the context of the PASS collaboration could be the starting point for more research in this area. The proposed project will be a collaboration between:

- The CN/ASD PAW and PIAF team: M.Ballintijn, R. Brun, O.Couet, F.Rademakers, J.Shiers
- Hewlett Packard (PIAF is a joint project with HP)
- The CN/PDP group members investigating Hierarchical Storage management: J.P. Baud, F.Hemmer, L.Robertson
- Physicists in the LHC experiments, NA48, NA49, including those designing the online systems.

3 ZOO

To avoid breaking encapsulation, distributed OO applications must deal only with object interfaces and should not care whether the object implementations are in the same process or on another machine halfway around the world. This ideal requires an object model that allows applications to transparently use both local and remote objects without sacrificing efficiency. Such an object model must address issues faced by all developers of distributed applications, providing a standard object programming interface that is not only system independent, but language independent as well. Even though an object reference identifies a particular object, it does not necessarily describe anything about the object's interface. Before an application can make use of an object, it must know what services the object provides. In CORBA, object interfaces are described in Interface Definition Language (IDL), a declarative language with a syntax resembling that of C++ . IDL provides basic data types (such as **short, long, float, double,** and **boolean**), constructed types (such as **struct** and **discriminated union**), and template types (such as **sequence** and **string**). These are used in operation declarations to define arguments types and return types. In turn, operations are used in interface declarations to define the services provided by objects.

We are proposing to develop a new system called ZOO to be compliant with the CORBA specifications. ZOO can be seen as, among other things, the replacement for the ZEBRA I/O systems FZ and RZ for Object Oriented Frameworks.

Several groups or individuals have expressed their interest in participating to the design and implementation of this system:

- R. Brun, A.Nathaniel, J.Shiers from CN/ASD
- Otto Schaile, OPAL group Freiburg University.
- Sunanda Banerjee (Bombay and CMS).
- Doug Olson (LBL) and several members of the STAR collaboration at RHIC: B.Greiman, C.Tull, P.Gee, D.Hall.

Support of Current Software Packages

It is our conviction that current software environment, as used for example by the LEP experiments, cannot be frozen now, as it will be used for the coming decade. Indeed LEP will continue to be the driving force in several areas.

Given that the responsibility of a good fraction of the people on these proposals includes providing software solutions for current experiments and those future experiments that will take data before the LHC, the overall approach is to provide staged solutions that will be used in the interim period before the LHC startup. In this way current and future experiments, including the LHC experiments themselves who are also using the current software, will benefit from incremental software advances which should lead to robust solutions at the time of the LHC startup. We see such a staged approach as being both realistic and practical and providing a set of inherent and useful milestones for all projects.

Relation to P55

We have carefully read the P55 proposal (CERN/DRDC/94-9 March 2, 1994) and believe that these proposals are in fact complementary to P55's stated goals and objectives. Indeed the P55 proposal makes reference to the ZOO and GEANT plans detailed here. We fully understand the need for close collaboration with P55. Futhermore we understand that in a formal sense interactions between software projects, both of a strictly R&D nature and those such as proposed here with definite products in mind, could be undertaken by the LHC Computing Review Board (LCRB).

References

[1] R.Brun, Software in HEP: A Strategy towards LHC, CN/DPG/94-7

[2] K.Amako, KEK. Presentation at CHEP94 is available under WWW http://kekux1.kek.jp/amako