SSC LABORATORY LIBRARY
JUN 2 9 1992

GEM TN-92-81

The Jazelle Data Manager for GEM

A.S. Johnson
Boston University

March 31, 1992

Abstract:

The Jazelle data management package has been developed over the proceeding seven years, initially for the SLD experiment at SLAC. The system was designed as a successor to earlier systems such as Bos and Zebra and includes many improvements over these systems such as self-documenting data structures, mnemonic access to all data, relational data structures, powerful machine-independent IO facilities and many mechanisms for presenting data to the physicist in an intuitive manner. This paper presents a brief and informal introduction to the JAZELLE data management system and highlights some of the ways in which its use could prove beneficial to the GEM Collaboration.

The Jazelle Data Manager for GEM

A.S. Johnson - Boston University

The Jazelle data management package has been developed over the proceeding seven years, initially for the SLD experiment at SLAC. The system was designed as a successor to earlier system such as Bos and Zebra and includes many improvements over these systems such as self-documenting data structures, mnemonic access to all data, relational data structures, powerful machine-independent IO facilities and many mechanisms for presenting data to the physicist in an intuitive manner. This paper presents a brief and informal introduction to the JAZELLE data management system and highlights some of the ways in which its use could prove beneficial to the GEM collaboration.

Advantages of Jazelle

Jazelle represents a new generation of data management system, gaining from experience with the many earlier tools used within HEP. Here we emphasis some of the ways in which Jazelle has progressed from those preceding systems.

Multi-Language Support

Jazelle provides support for multiple languages. This allows data to be accessed from and shared between programs written in any of the supported languages. Currently supported languages include C and Fortran 77. Work is in progress on Fortran 90 support. Although it is still anticipated that a single language will be chosen for most of the collaboration's software, the increasing use of languages such as C and C++ in real-time and GUI applications make it increasingly likely that there will be at least some software written in another language.

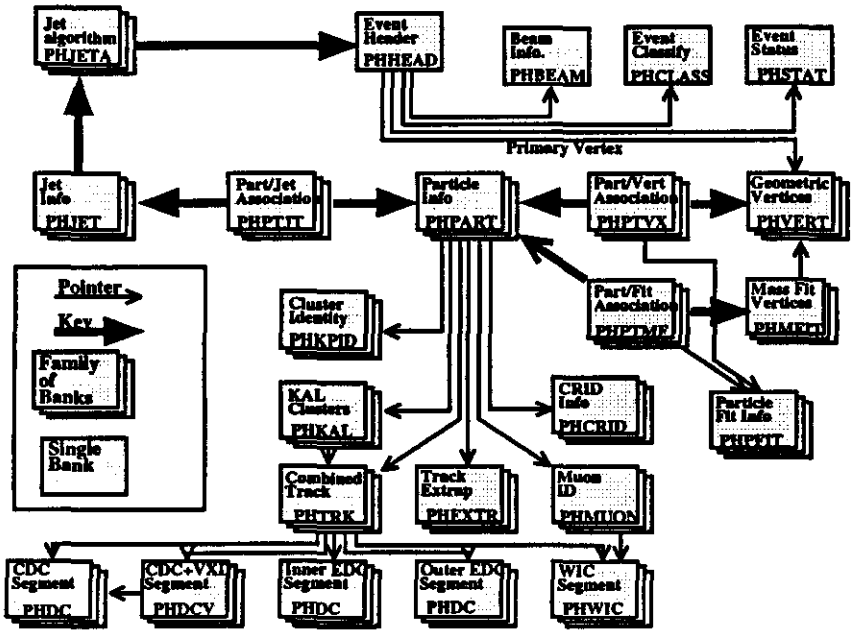

Figure 1 A typical data structure build with Jazelle.

The SLD data structure is an example of a complex data structure build using Jazelle, using both hiearchical and relational relationships.

Simple Mnemonic Access to Data

Jazelle allows clear mnemonic access to data from any of the supported languages. Data is accessed using expressions of the form:

```
Track%Ptot

Track%Hit(i).x
```

This is a huge improvement over earlier data management systems such as Zebra in which data are accessed using cryptic expressions such as:

```
Q(Ltr(Track)+12+I*17)
```

The use of Zebra results in code that is difficult to read and understand and which is prone to errors. By comparison, code using Jazelle is easy to read and Jazelle takes care of many of the chores that lead to hard to find errors using other systems (such as writing IQ instead of Q in the above expression).

For languages which do not support data structures (e.g. F77) Jazelle uses a simple pre-processor to translate the expression into native language expressions. In all cases the changes made by the pre-processor are limited to the expression in question, and are fully com-

mented so that the pre-processor has minimal impact on the ability to debug the code. In languages which support data structures natively (eg C,F90) no pre-processor is required.

Self Documenting Data Description Language

The structure of each Jazelle bank is described in a human readable, self-documenting template file. Each data element is typed (real, integer etc.), named and described there. Banks may contain arbitrary combinations of data types in any order, including both scalar and vector types. In addition to the more common data types, Jazelle provides support for often used HEP data types such as BITS, PARTID (particle id) and TIME. Jazelle can be extended with experiment specific data types.

Powerful Tools for Access to Data

Jazelle contains many utilities for presenting data to the user in a variety of different formats and levels of detail. Jazelle includes a complete set of interactive commands which enable users to manipulate and view data structures, including tools with which users can build their own arbitrary views of the data. These tools have been interfaced to various debuggers and interactive data analysis systems. The combination of Jazelle plus an interactive event display or analysis system (such as PAW or IDA) produces a very productive physics analysis environment.

A recent improvement is the addition of an X/Motif based data browser which allows data structures to be viewed and manipulated using a fully interactive GUI. The Jazelle data browser can be used as a standalone program or as part of some larger system such as an online data collection system, or a physics analysis tool.

Relational Data Model

Jazelle provides facilities for describing data structures using a relational technique, in additional to the more common hierarchical data descriptions found in earlier data managers. This provides a natural way of describing the type of NxM relationships often found in HEP data, such as the relationship between tracks and vertices when ambiguities are included.

Virtual Memory Usage

Jazelle uses the virtual memory services of whatever machine it is running on. This eliminates the problems of garbage collection and running out of space in the large pre-assigned common block used by earlier systems and makes optimal use of the host machines resources and capabilities.

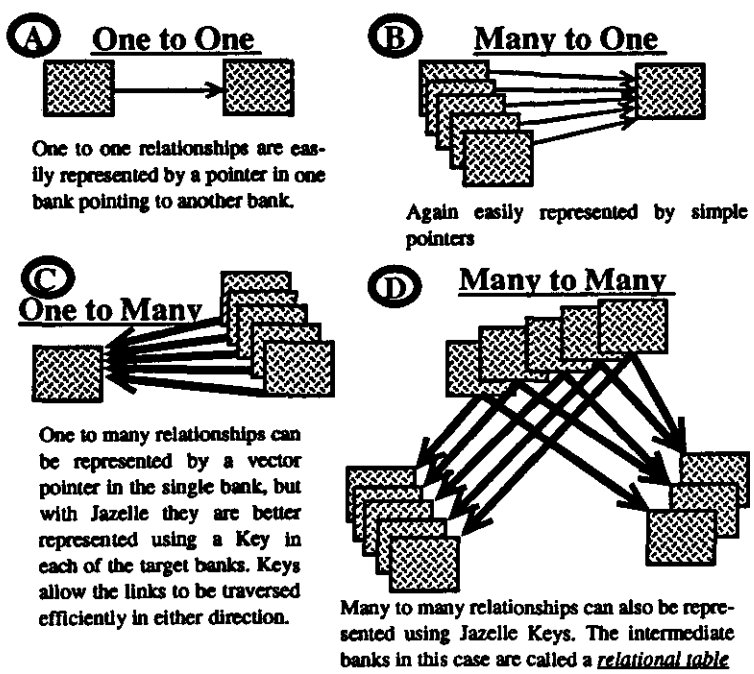

Figure 2 Relationships supported naturally by Jazelle

Tried and Tested

Jazelle has been used by SLD for over 5 years to handle all data storage and transport from the online VAX to the end of the reconstruction and physics analysis chain. Jazelle is also used to manage calibration constants, detector geometry, physics parameters and program control parameters.

Many practical problems associated with transporting and storing large data samples in a real HEP experiment have been encountered and overcome during this period.

Multiple System Support

Jazelle has run for many years on both VM and VMS machines, providing full functionality on both platforms. Most capabilities are now avail-

able under Unix and full IO capability should be available soon.

Powerful IO System

Jazelle includes facilities to handle input and output of data to both sequential and indexed (random access) files. Data is written to disk or tape in the native format of the host machine, but is translated transparently when read on a different machine. Jazelle supports multiple IO protocols to allow efficient IO of structures ranging from a few bytes, to many Megabytes. Jazelle also provides support for asynchronous inter-process IO.

Common Myths about Data Management

The use of C would eliminate the need for a data management system

In the past data managers have sometimes been viewed as only necessary to compensate for the inadequacies of older languages such as Fortran 77. In reality although newer languages such as C have powerful facilities for creating and manipulating data structures within programs, none of them have the tools that are needed by HEP collaborations to transport complex data structures between programs or between machines.

Jazelle provides many essential tools such as machine independent IO and the ability to view and manipulate data interactively that native C can not provide. At the same time Jazelle allows C's native

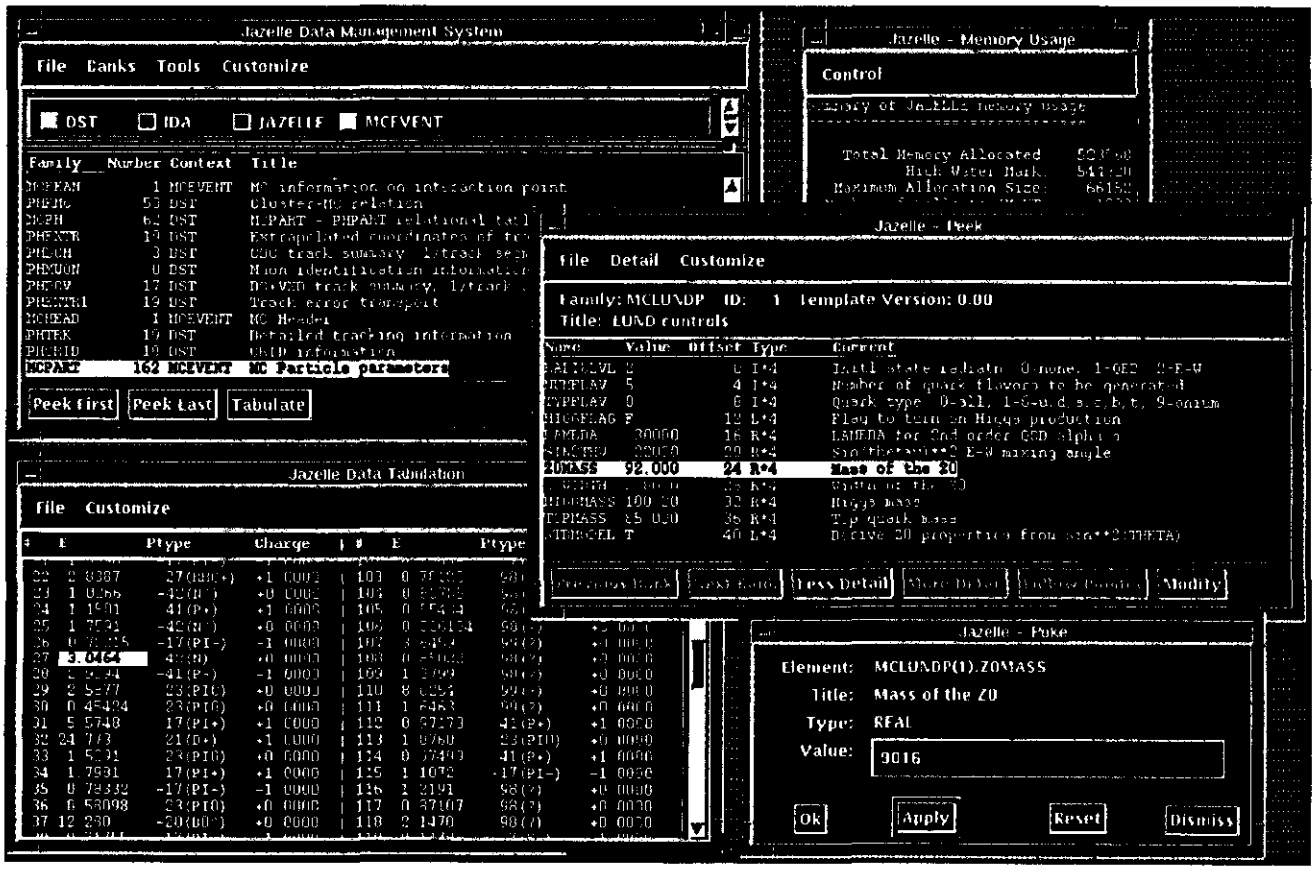

Figure 3 The Jazelle Data Browser

The Jazelle Data Browser is an X/Motif based data browser that can be used standalone or as part of a data analysis or online system. The browser allows users to examine and, if appropriate, modify data in Jazelle banks.

handling of data structures to be exploited to write easy to understand programs.

The use of Fortran 90 would eliminate the need for a data management system

The reality here is similar to the C case above, although in some ways worse. In Fortran 90 it is illegal to write out a structure that contains a pointer, and it is impossible to have pointers that can be associated to more than one type of object. This means that not only is Fortran 90 alone unable to provide the support for data structures needed by HEP, but it makes it impossible to write the kind of data manager needed for HEP without violating the standard.

In practice it will be possible to "cheat" as HEP always has in using Fortran, although until several real implementations of F90 are available it is hard to say exactly how.

Using Jazelle it is possible to use Fortran 90-like structures now with Fortran 77, and to switch to using Fortran 90 later. A pre-processor would be used with Jazelle to translate F90 data references into F77. When F90 was available to all collaborators the preprocessor would be dropped, and Jazelle would be used only for IO and for its data viewing tools.

Data management systems are easy to write and GEM should create its own

I spread this myth when I was starting to write Jazelle for SLD. It was a lie; although it is possible to produce a semi-working system quickly, the work needed to bring the system to completion and to handle the IO with sufficient speed and flexibility is substantial. The current Jazelle system represents at least 10 man years of effort.

Another important consideration is the effect that an unstable data management system has on the moral of people trying to use it to write reconstruction code.

Using GEANT forces the use of Zebra

The SLD collaboration has used GEANT extensively as the basis of its full MC. Although Zebra is used internally to Geant, the interface that all physicists see is based entirely on Jazelle. Only one or two people in the entire collaboration need to know the details of the interface, the rest of the collaboration is spared the need to use or understand Zebra.

Further Information

The following publications describe Jazelle in more detail:

1. A.S.Johnson et al., "*Jazelle: An enhanced Data Management System for High Energy Physics*", AIP conference proceedings **209**, 285. Proceedings of the 8th "Computing in High Energy Physics" conference, Santa Fe, New Mexico, April 1990(*).
2. A.S.Johnson et al. , "*Jazelle: History, Status and Future Plans*", proceedings of the 14th workshop of the INFN Eloisatron Project, Erice, Sicily, Italy, November 1990(*).
3. T Hansl-Kozanecka, "*A Comparison of Data Management Systems used in High Energy Physics*", proceedings of the 14th workshop of the INFN Eloisatron Project, Erice, Sicily, Italy, November 1990.
4. A.S.Johnson and D.J.Sherden, "*Jazelle Users Manual*", SLAC-PUB-263(*).

Those publications marked (*) are also available as postscript files by anonymous FTP from He-*plib.Slac.Stanford.Edu in directory /pub/Jazelle.*

In addition you are welcome to contact the author with any questions of comments: Tony Johnson (TonyJ@Slacvx.Slac.Stanford.Edu); Phone: (415) 926 2278; Address: Mail Stop 71, SLAC, Stanford CA 94309.