SUMMARY AND CONCLUSIONS OF THE IADBG WORKSHOP ON DATABASES FOR ACCELERATORS Argonne National Laboratory, 6-8th November 1995

J. Poole
CERN, Geneva, Switzerland

1 Introduction

The IADBG Workshop on Databases for Accelerators was held in the new conference centre at Argonne National Laboratory and was attended by over 40 people (more than 12% of the ICALEPCS attendance). The workshop consisted of 3 major sessions and a summary session on the final day. The workshop programme was as follows:

Part I - Functionality and Data

Session Chairman: Ian MacGregor (SLAC)

CERN, Josi Schinzel
SLAC, George Crane
FNAL, Kelley Trombly-Freytag and Tim Zingelman
BNL, Garry Trahern
CEBAF, Sue Witherspoon

Part II - Tools and Applications

Session Chairman: Roger Bailey (CERN)

Oracle at CERN, Pierre Strubin
Oracle at SLAC, Andrea Chan
Sybase at RHIC, Garry Trahern
Sybase at FNAL, Kelley Trombly-Freytag and Tim Zingelman
ObjectStore at CEBAF, Matt Bickley

Part III - Techniques and Problem Areas

Session Chairman: Chip Watson (CEBAF)

Methodologies, Josi Schinzel
On-line databases, Mike Lamont
Very large databases, Ronnie Billen
Data analysis and visualisation, open discussion

Part IV - Summary and Conclusions

Session Chairman: John Poole (CERN)

Summary of Functionality and Data session, Ian MacGregor
Summary of Tools and Applications session, Roger Bailey
Summary of Techniques and Problem Areas session, Chip Watson
Open discussion to define wish list, sharing areas, further studies, future workshops, working groups …

A summary of the important issues and conclusions is given below. The proceedings are available electronically through the IADBG WWW pages[1].

[1] The IADBG home page URL is http://www.cern.ch/IADBG/Welcome.html

2 Functionality and Data

In the first few talks it became clear that a wide variety of database management systems were being used and that every conceivable type of data in accelerator laboratories was concerned. The data being handled falls into the following broad areas:

- Laboratory management – personnel, finance, pay roll, documentation, project management
- Equipment management – construction, maintenance
- Control – off-line reference data and on-line active data
- Beam related data management – measurement and physics data

Some interesting developments in the field of geographical data and relational representation of objects were reported from Brookhaven National Laboratory.

If one considers what are generally referred to as database activities, then memory resident home-made solutions dominate. In the relational field, Oracle and Sybase dominate. The only object database discussed in detail at the workshop was ObjectStore, which is being used at CEBAF.

The diverse choice of solutions to the data management problem arises because there is no unique alternative. Databases have been built to suit particular problems or users and even where a coherent solution has been adopted for a project, it is necessary to have partitioned implementations to accommodate the conflicting requirements (high performance query engines, development systems etc). In most cases reported there is, or will be, more than 1 type of implementation e.g. relational and memory resident or object and memory resident.

The diversity itself leads to maintenance problems: expertise is required in several areas and replacement of legacy systems becomes increasingly difficult. This is probably an indication that the field is still in its infancy. Some ingenious implementations, linking heterogeneous systems were presented (BNL, SLAC). These are necessary to enable activities which require input from several disciplines.

3 Tools and Applications

Tools are used for data manipulation and the range which is used for data input is rather limited. For relational systems and expert users, data input is often done using SQL. Whilst this is very easy to use, it is also very easy to corrupt or destroy data and this is why it is restricted to *experts* (several of whom admitted to making serious errors this way). Oracle provides a graphical interface (Oracle*Forms) which allows the user to interact in a very controlled way with the database but at the expense of development effort.

In the PC world it is possible to connect to relational databases through commercial products like EXCEL and several instances of such applications were reported. Apart from these few tools there are numerous home made interfaces which have been built to enable data input.

The variety of tools available for data extraction is far greater and includes proprietary interfaces from Oracle (Oracle*Browser, Oracle*Report, Oracle*Graphics …), EXCEL, PVWAVE, tcl/tk etc. Whilst there is a great variety there was a general feeling that many of these products are not very popular with the users. In general, the public domain tools like tcl/tk and WWW were more popular and users were happier with them - perhaps because they have a more familiar look and feel and are not user hostile.

In the case of ObjectStore the only way to interact with the database is through C++ and it requires a knowledge of the structure of the database in order to save or retrieve data.

Applications have been built using all of the tools mentioned above as well as programs written using programmatic interfaces to the database. All of the databases discussed at the workshop are delivered with a set of libraries which allow developers to interact with the database from within a program.

It was emphasised that the majority of the work associated with database systems is not creating the structures and maintaining the data, but building and maintaining the applications. The possibility of enforcing referential integrity and constraints within the database (which arrived with the latest releases of the relational DBMS's) was seen as a big step forward. This development removes the necessity of building in controls to enforce referential integrity and constraints in every application. However, this represents only a small part of the total effort required for application development.

It was agreed that the area which holds the greatest promise for application development is that of the WWW interfaces.

SLAC and BNL both reported developments, both based on in-house developments using public domain tools.

4 Methods

Experience at CERN with project management using the European Space Agency (ESA) standards was discussed. The standard was succinctly described as 'rigorously applied common sense', but with the warning that one should always apply methods in a pragmatic way.

From the review of information technology methods the main conclusion was that it is essential to follow some methodology but that the choice seems to be largely a matter of taste. The group of people who actually design and build the databases is a small fraction of the people involved in database activities ($<33\%$) and most of them were following some methodology and using tools. A number of tools were described:

- Teamwork, which supports Structured Analysis (SA), Structured Design (SD), Entity Relationship Modeling (ERM) and Objects (next/latest release ?)
- Oracle*CASE, which supports ERM and functional decomposition
- ERDraw, which supports ERM
- RIDL, which supported the Natural Information Analysis Method (NIAM) but is no longer supported commercially
- Object Protocol Method (OPM)
- Select OMT, which supports the Rumbaugh method
- TOT, an in-house tool at FNAL

The difficulties of learning a methodology *AND* tool were discussed in some length. It was generally agreed that there is a considerable investment to be made and there is a danger that new users will treat the method like a religion. All of the delegates who had been through the experience agreed that the discipline of a methodology is a vital ingredient in the production of a good implementation. It was also agreed that for *smaller* projects a complete analysis and design, using formal methods, was an overkill.

5 Techniques

5.1 On-line Control Databases

The majority of such implementations are built around what can be classified as memory resident custom-built databases. The case of the LEP on-line database was examined in some detail. This has been built using Oracle and is in the control path: control applications write settings to the database and the process connected to the hardware receives a signal which causes it to get the new settings from the database and then it changes the hardware. Client applications use the database to get data, rather than polling the equipment. This development has been possible because of the latency in the LEP control system, due in part to its size and in part by choice, where changing hardware settings should be done in a time <1s.

Other large control systems which use memory resident systems (EPICS, SLC) often have much tighter constraints on response times. The performance limits for a relational database in a control system have not yet been determined.

5.2 Very Large Databases

When then LEP logging database was conceived a few years ago it was considered to be a very large database ($\sim$10 Gbytes), whereas today people are discussing volumes of data orders of magnitude larger. However, in terms of the database systems in use at accelerator laboratories it remains one of the largest to have been implemented in a relational system. Tables with up to a million rows are in use and several special techniques had to be employed in order to make the system perform adequately.

It was concluded that using an RDBMS for such a system is a reasonable solution as long as the I/O requirements are reasonable. In the LEP case around 120 Mbytes per day are being written to the database and this is some (unspecified) way below the limit.

6 Problem Areas

6.1 Data Visualisation and Analysis

Many tools are being used to present data to the user and for analysis (e.g. EXCEL, XRTgraph, PAW, PVWave …). In general the mature sites were relatively happy with the facilities available. A new in-house system which is being used extensively at the APS at Argonne is the SDDS (Self Describing Data Set) Toolkit. This uses the custom files in combination with a powerful scripting language which allows data analysis, reduction and visualisation. Whilst the system is very performant it was felt that the lack of data management was a distinct disadvantage.

A new system is being designed for CEBAF and this is based on an integrated set of tools which will allow control of equipment to be integrated with the data gathering, analysis and display.

7 Future Plans

It was agreed that the IADBG should start a number of new activities in order to help members in the future. The first idea was to establish a mail exploder so that members can send requests for help and/or advice on database problems. This service will supplement the E-mail newsletters because not all members will necessarily wish to subscribe to the exploder.

It was felt that some more detailed description of database implementations would be of great use to people starting new developments. More detail than was presented at the workshop is needed, but the data structure definitions would be too detailed and not particularly useful – some short descriptive paragraphs were thought to be most useful. The information will be made available through the IADBG WWW pages.

The success enjoyed by WWW applications giving access to database information was stressed many times during the workshop. It was therefore agreed that examples of such implementations (the code) will also be made available through the IADBG WWW pages.

On a number of occasions there were discussions about the performance capabilities of various database systems. There are indications that for some applications object oriented databases can considerably out perform relational implementations but it was clear that there was no solid basis for comparison. It was decided that some kind of benchmark test will be defined so that meaningful comparisons can be made. The test should be something which is relevant to accelerator projects and which can serve as a guide to the relative performance of object and relational databases, for example.

The IADBG was started in 1994 and had held meetings at EPAC94 and PAC95 before the workshop. It was agreed that the group has now established itself on a solid basis and that it should have at least one co-ordinator on each continent. J. Poole agreed to continue looking after Europe and A. Chan (SLAC) agreed to look after N. America.

Finally it was unanimously agreed that the workshop had been extremely useful and that it would be appropriate to repeat it in the future. It was felt that in about 2 years it would be useful but that the situation will be reviewed at future meetings of the IADBG. The next meeting will be at the EPAC[2], which will be held in Sitges, Barcelona, Spain from 10 - 14th June 1996.

8 Acknowledgement

I would like to express my thanks to all of the speakers and session chairmen who worked hard and made the workshop such a success. I must also thank the organising committees of ICALEPCS for their help with the organisation of the workshop. Finally I would like to thank Bill McDowell of Argonne National Laboratory for arranging the meeting room, refreshments and on-site transport.

[2] EPAC WWW pages are accessible through http://www.cern.ch/EPAC/Welcome.html