EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN/LHCC 96-15
LCRB Status Report/RD45
22 February 1996

RD45 - A PERSISTENT OBJECT MANAGER FOR HEP

Edward May, David Malon
Argonne National Laboratory
Argonne Illinois USA

Ryszard Zybert
School of Physics and Space Research
University of Birmingham, UK

Jacek Becla, Pavel Binko Dirk Duellmann, Bernardino Ferrero Merlino, Gunter Folger, Jamie Shiers (spokesman)
CERN/CN-ASD
Geneva, Switzerland

Michel Hansroul, Lassi Tuura, Ian Willers
CERN/ECP
Geneva Switzerland

Otto Schaile
Freiburg University
Freiburg Germany

Predrag Buncic, Martin Pußchke
GSI Darmstadt Germany

Boris Klochkov
IHEP Protvino Moscow Region Russia

Chris Day David Quarrie
Lawrence Berkeley National Laboratory
Berkeley, CA USA

Youhei Morita
KEK, Oho Tsukuba Ibaraki 305 Japan

Aneta Baran, Staszek Jagielski, Andrzej Sobala, Witold Wajda
Institute of Nuclear Physics
Krakow, Poland

Vincenzo Innocente
INFN Sezione di Napoli Napoli Italy

RD Schaffer
Laboratoire de l Accelerateur Lineaire
Orsay France

Giovanni Organtini
Universita' di Roma ' La Sapienza' and Istituto Nazionale di Fisica Nucleare - Sez di Roma
P le Aldo Moro, 2 - 00185 ROMA Italy

Sunanda Banerjee
Tata Institute of Fundamental Research Bombay, India

Nobuhiko Katayama
Laboratory of Nuclear Study, Wilson Lab, Cornell University, Ithaca New York, 14853, U S A

Anwarul Hasan
ETH Zurich, Switzerland and University of Cyprus, Cyprus

(The list of RD45 friends may be accessed via the URL http //wwwcn cern ch/pl/cernlib/rd45/people/friends html)

1 Executive summary

The RD45 project, which was approved in February 1995, is investigating solutions to the problem of object persistence (frequently referred to as the "I/O" problem), for LHC-era HEP experiments The project, which has placed strong emphasis on understanding whether standard, commercial solutions can be used as components in the eventual solution, believes that it has identified a viable solution, which is outlined below

1 1 Proposed solution

We believe that the combination of commercial, standards-conforming Object Databases together with commercial, standards-conforming Mass Storage Systems can be used as key building blocks in a powerful, scalable, affordable solution to the problem of handling persistent objects for LHC-era experiments This solution is open to many computing model scenarios, and is also capable of handling the inevitable evolution both in technology, and in the computing model itself, that will occur throughout the lifetime of an LHC experiment

1 2 Summary of activities during the first year

The results of the first year of the RD45 project can be summarised as follows

- identification of the key standards and technologies involved,
- a clear focus on standard, commercial solutions,
- participation in the appropriate standards bodies,
- extensive training in these key areas,
- evaluation of a number of potential solutions, based upon interim system requirements,
- the identification of a powerful solution to the object persistence problem - a scheme whereby multiple PB of data can be handled without stressing the available technologies,
- close contacts with the appropriate vendors,
- successful prototyping of storage and retrieval (with encouraging performance) of HEP event data,
- wide exposure and general approval of the proposed solution within the community

1 3 Conclusions

- We have identified a powerful and scalable solution to the problem of handling persistent objects in the multi-PB (10^{15} bytes) region, in a fully-distributed, heterogeneous environment,
- Results from initial prototypes have been very encouraging, and we have identified a number of opportunities for the production deployment of elements of the proposed solution in HEP experiments starting from 1996,
- Much research still remains to be done, particularly on the overall computing and object models, and in understanding how to optimise data placement for the most efficient access However, we are confident that, together with the experiments themselves, we can make significant progress in these areas during the coming year

2 Milestones for the first year

RD45 was presented to the CERN DRDC in September 1994 and approved in February 1995, for an initial period of one year, with the following milestones

- [produce] *a requirements specification for the management of persistent objects typical of HEP data together with criteria for evaluating potential implementations*
- [perform] *an evaluation of the suitability of ODMG's Object Definition Language for specifying an object model describing HEP event data*
- *Starting from such a model, a prototype using commercial ODBMSes that conform to the ODMG standard* [should be developed] *The functionality and performance of the ODBMSes should be evaluated*

It should be noted that the milestones concentrate on event data Studies or prototypes based on other HEP data should not be excluded, especially if they are valuable to gain experience in the initial months

3 Interim Status Report

At the time of the LCRB meeting in November 1995, a presentation on the status of RD45 was made The main reasons for making such a report were

- we believed that we had identified a solution to the persistence problem that used commercial solutions - Object Databases together with Mass Storage Systems, and that it was important to report on this change of focus of the project,
- the RD45 collaboration was still relatively small, and largely concentrated at CERN We felt that it was important to make additional publicity for the project, to increase awareness of its activities and to attract additional manpower

The referees comments on this interim report are reproduced below

3 1 RD45 Successes

- *Rapid and successful focus on standards and commercial ODBMSes,*
- *Identification of federated databases as a valid solution for high volume event data,*
- *Very promising tests of event data retrieval from a commercial ODBMS,*
- *Modest expansion of the collaboration to include some key experts in HEP data management*

3 2 RD45 Problems

- *RD45 is still a small collaboration Most of the work is being done by CERN/CN members (although many other people are taking advantage of this work),*

- *ATLAS participation is urgently needed,*
- *Neither CMS nor ATLAS is close to being able to help RD45 produce a requirements specification (the first milestone),*
- *RD45 needs to find an experiment willing to try an ODBMS-based physics analysis in 1996 rather than 2004*

3 3 RD45 Recommendations

Complement to the First Milestone

- *Produce a 'Statement of Probable Capabilities' for a HEP persistent object management system based on commercial ODBMSes and large-market mass storage systems*

3 4 Progress since interim report

Since the interim report in November 1995, progress has been made in a number of areas In particular, as well as completing the revised milestones, we have made significant progress in all of the problem areas identified by the interim review Full details are given below

4 Organisation

The RD45 project was originally largely centralised at CERN, with the exception of a subgroup in Krakow in Poland and a collaborator at Cornell University Following CHEP '95, two people from the PASS collaboration have joined, based at Argonne National Laboratory in the US More recently, the collaboration has grown with members from BaBar (also ex-PASS collaborators) and DESY Furthermore, we have established close contacts with the SDSS project at FNAL, which is also focusing on Objectivity/DB for object persistence

Information on the RD45 project is made available through the World Wide Web, and through AFS Almost all of the information is publicly accessible - only licensed material, such as commercial software or information obtained under non-disclosure agreements, is protected Regular bulletins are sent by electronic mail to all project members as well as to "friends" of the project Meetings are held primarily at CERN, and the project has also been widely presented both in and outside CERN - numerous presentations have been made to sub-groups of ATLAS, CMS and RD41 (MOOSE) as well as three sessions at the OO R&D day in November 1995 and presentations to ALICE and LHC-B are scheduled for the coming weeks Additionally, presentations have been made in DESY, Krakow, Prague and Heidelberg as well as at a BaBar collaboration meeting and at the 14th IEEE symposium on Mass Storage Systems and CHEP '95 A "birds-of-a-feather" session on persistent object management was organised at CHEP '95 and was widely attended by people from US institutes and FNAL in particular A workshop is scheduled for mid-March in KEK, and an Objectivity workshop took place at CERN in February 1996

5 Relationships with other Projects

5 1 BaBar

Following a number of discussions, which started during CHEP '95 and then continued at a BaBar collaboration meeting in Paris and finally at CERN, it has been agreed that BaBar and RD45 should work closely together Two members of the BaBar collaboration have now joined RD45, and it is hoped that others might follow Very similar programs of work for 1996 have been envisaged by the two sides, and a close collaboration will clearly be profitable

5 2 CAP

The Computing for Analysis (CAP) project[1] is aimed at providing improved platforms and software for performing HEP experiment analysis in the post-event-reconstruction stages The current focus is on "data mining[2]" applications which must rapidly scan through vast quantities of physics data Although the CAP project has adopted a rather different strategy to that of RD45, namely the use of a non-standard conforming object manager and query language, many of the requirements that we are trying to address are common We have, therefore, established a communication channel between the two projects

5 3 DESY

Following a recent workshop at DESY, it was agreed that DESY would actively participate in RD45-related activities in the future The following items were discussed

- organisation of an Objectivity/DB course at DESY,
- participation in the Objectivity workshop at CERN in February,
- design and implementation of an "event-directory" (tagDB) for the ZEUS experiment,
- testing of Objectivity/DB with SGI's 64-bit filesystem (including issues related to very large databases/files),
- integration and testing of Objectivity/DB with the Open Storage Manager mass storage system

This workplan clearly fits in extremely well with the overall plan for RD45

5 4 GEANT-4

The GEANT-4 (RD44) project[3] is an R&D project whose goal is to develop an object-oriented toolkit for simulation in HEP As described below (see section 14 11), the first version of GEANT-4 should have persistence as provided by RD45 There are close contacts between the two projects, and common solutions have been adopted where-ever possible (e g for code management, choice of class

1 See http://fnhppc.fnal.gov/cap/cap.html

2 Data mining is usually defined as being "the efficient discovery of previously unknown relationships and global patterns in large databases"

3 See http://wwwcn.cern.ch/pl/geant/geant4.html

libraries for prototyping etc) As these common solutions have already been reported upon in the GEANT-4 status report[4], we do not discuss them here

5 5 HPSS

The High Performance Storage System Project (HPSS)[5], is a project based at the National Storage Laboratory (NSL) in the US, aimed at producing a next-generation mass storage system The development partners include the US national laboratories (LANL, LLNL etc) and IBM Sites with HPSS deployment plans include Argonne National Laboratory, Cornell Theory Center and FNAL Although there are no formal contacts between RD45 and HPSS, we receive regular reports regarding the progress of HPSS

5 6 PASS

The Petabyte Access and Storage Solutions (PASS)[6] project lead the way in investigating how Object Databases could be used to store and manage HEP data [14][16][17] The project, which was originally oriented towards the experiments at the SSCL, has now terminated However, the key ideas of the PASS project are compatible with those of RD45, and a number of the members of the PASS project have now joined RD45

5 7 SDSS

The Sloan Digital Sky Survey (SDSS[7]) is a project involving a number of sites in the US, including FNAL, that is using ODBMS technology and Objectivity/DB in particular The goal of the project is to build and operate a dedicated telescope suitable for wide-angle surveys of the sky which will address critical issues in extra-galactic astronomy, especially in the field of large-scale structure Information on experience with Objectivity/DB is freely exchanged between the two projects[8]

6 Requirements for a HEP Persistent Object Management System

In order to help us in evaluating potential solutions to the persistency problem, we established a number of requirements, some of which were listed in the original project proposal We stress that these requirements are preliminary, and will be revised according to the evolving LHC computing models

Although these requirements are very high-level, they already eliminate many of the possible solutions to the problem For example, some solutions are tied to a single architecture (number of bits/word, byte ordering, data format, and/or operating system) and hence can be immediately ruled out

4 See http://asdwww.cern.ch/pl/geant/documents/StatusReport95.ps

5 See http://www.llnl.gov/liv_comp/nsl/hpss/hpss.html

6 See http://sun2.hep.anl.gov

7 See http://www-sdss.fnal.gov:8000/

8 An example of such an exchange of information is the performance evaluation of Objectivity/DB s C++ and SQL++ API s performed in the context of the SDSS project and reported on below (see section 9 12)

Others do not support the full C++ Object Model and are therefore also eliminated In particular, we stress that neither language extensions, such as E or O++, nor light-weight object managers, such as Ptool, are able to satisfy these preliminary requirements

6 1 Object Manager Requirements

The Object Manager must

- not impose any arbitrary restrictions on the Object Model of the collaborations For example, the full C++ Object Model must be supported by the Object Manager,
- work in the fully-distributed, heterogeneous environment,
- appear to the user as a single, logical system, even if composed of multiple physical instances running on a variety of platforms,
- be capable of supporting a variety of different computing models, perhaps even concurrently, and be capable of supporting the evolution of such models,
- be capable of providing access to the services and/or data of individual objects in a single, logical multi-PB system

The only systems that we know of capable of satisfying this initial list of requirements are full Object Databases However, in the following, one may freely replace "Object Database" by "Object Manager/Store"

6 2 Object Database Requirements

A number of these requirements were obtained from the Capacity and Scalability section of the report "DBMS Needs Assessment for Objects" [2] This report may also be used as the basis for a much more complete set of requirements than can be listed here

The Object Database (ODBMS) must

- support all 11 mandatory requirements listed in "The Object-Oriented Database Manifesto" [3],
- comply to the bindings defined by the Object Database Management Group (ODMG) [11] in terms of Object Model, C++ and ODL bindings[9],
- provide replication of user data,
- support schema evolution and object instance migration,
- be capable of supporting and exploiting parallelism, e g parallel load, parallel query etc ,
- provide a mechanism for transparently integrating the ODBMS with a Mass Storage System that is site-configurable,
- support individual physical databases of at least 100 GB by 1997, and individual databases of at least 1 TB by 2000[10],

9 We do not list OQL compliance as a requirement at this stage, pending the outcome of the attempts to align OQL with SQL3 In the future, a Java binding may also become a requirement, although work on this is only just starting within the ODMG

10 This requirement, and also the bitfile size that must be supported by the storage system, come from the model of the proposed solution, whereby physical databases are limited to 100 GB - 1 TB See section 8 for more details

- provide a mechanism whereby a single, logical database can be build out of multiple physical databases, distributed across multiple, possibly heterogeneous, database/file-servers spread across a wide area network At least 2^{16} physical databases per logical database must be supported in 1997 and at least 2^{32} by 2000,
- impose no arbitrary limits on the number of classes, number of attributes per class or the number of instances of individual classes Should such architectural limits exist, then they should in no case be less than 2^{32} - a higher limit, such as 2^{64} being strongly preferred,
- impose no arbitrary limits on the number of simultaneous clients Should such limits exist, then they should exceed 2^{16},
- offer scalable (i e constant or linear) performance characteristics,
- represent a "small" (e g 10%) overhead related to the raw performance offered by the underlying filesystem or raw devices

6 3 Storage System Requirements

Whilst remaining relatively optimistic regarding the possibility of multi-PB of disk space by the start-up of LHC, we nevertheless include a requirement for a mass storage system

The Storage System must

- support individual bitfiles of at least 1 TB,
- be capable of scaling to the multi-PB (10 PB in 2004) region A solution that used a number of separate storage systems but that was nevertheless able to appear as a single system, would be acceptable,
- support data rates of at least 100 MB/second per stream,
- provide a total sustainable through-put of at least 1 GB/second,
- allow a "class of service" to be associated with different categories of bitfiles, which would determine the migration of the relevant bitfiles through the storage hierarchy,
- support both a standard API (e g the X/Open draft XBSA, POSIX file I/O, the DMIG DMAPI) and filesystem interface (e g AFS/DFS, VFS),
- support a variety of industry standard media formats,
- be capable of transparently migrating user data from obsolete media to newer media,
- support high-level clustering of bitfiles[11]

6 4 Financial and Manpower Constraints

- Any proposed solution must be affordable, both in terms of the manpower and financial terms, not just for large data-centres but also for smaller collaborating institutes

11 Arguably, this feature is unnecessary if the "class of service" feature if sufficiently well implemented

7 Selection of an ODBMS for Prototyping Activities

Although we did not perform a full evaluation of any ODBMS, we obtained either trial or full licenses for O_2, Objectivity/DB, Objectstore and POET - this preliminary selection being based upon a number of ODBMS evaluations Our strategy has been to identify the most suitable product for prototyping activities, and to postpone any decision regarding an eventual acquisition until much closer to the start of LHC data-taking, e g around 2000 In order to identify the best product for prototyping, we contacted all vendors taking part in the ODMG effort, and invited them to make presentations of their products at CERN Surprisingly, not all vendors replied - initially, only O_2, Objectstore and Objectivity replied, although Matisse, Poet, Versant and UniSQL have since responded

To aid our selection, we consulted all freely available comparisons of Object Databases We also made heavy use of "DBMS Needs Assessment for Objects" [2], which compares the major databases according to a wide range of criteria

O_2, Objectstore and Objectivity/DB all rate highly in independent database evaluations, and hence we were reasonably confident of our initial selection Based on the presentations made at CERN and on subsequent discussions, we narrowed this selection to two[12] - O_2 and Objectivity/DB Finally, having followed courses on both of these two databases, and having obtained the software and documentation, our focus has been on Objectivity/DB

7 1 Objectivity/DB

Objectivity/DB offers high-performance, scalable, robust database support for systems with many users and objects spread across multiple databases, servers, networks, and languages As is implied in the above, the company focuses strongly on high-end applications, and thus appears to be a good match for our requirements The product is available for a wide range of Unix platforms, VMS and there are also various PC versions including NT and Windows '95 The documentation, which is also available online, is very professional and our experience with the support that the company offers has been excellent In the current release, Objectivity/DB does not fully comply to all of the bindings specified by the ODMG However, the areas in which it does not comply correspond to those where the standard is not fully worked out For example, the current (V1 2) ODMG C++ binding does not conform to the Standard Template Library part of the draft C++ standard, although it is the intent of the ODMG that V2 0 of the ODMG standard will be fully consistent with the C++ standard in this respect Similarly, discussions are on-going between the OQL (ODMG) and SQL3 (X3H2) groups, regarding convergence between these two standards

7 2 O_2

O_2 is an ODBMS that was developed in the Altair project [4], which was a five-year project, starting in September 1986, with the target of designing and implementing a next-generation database system The system is now a product, maintained and marketed by O_2 Technology

12 POET was not considered further as it does not appear to be a high-end product Objectstore has neither a scalable architecture, nor were the company willing to work with us

Our main concerns with O_2 are

- no support for ODMG's ODL - schema are defined using the O_2C language[13] (which, unlike ODL, is not based upon C++ and is confusing for the C++ user),
- no support (yet) for heterogeneity,
- poor documentation

In addition, we have been unable to obtain information regarding their plans for fully distributed environments, nor for handling very large databases[14]

It is our conclusion that O_2 is best suited to projects with relatively small amounts of data and number of users In its current state, we cannot see how it could be used to handle multiple PB of data and hundreds of concurrent users

7 3 Conclusions

It became increasingly clear that Objectivity/DB had a number of significant advantages over the other databases, in terms of meeting our preliminary requirements Examples of these advantages include

- the most scalable architecture, in terms of size of database, number of clients and database servers etc ,
- existing deployments of very large databases, including sites where the database was used in conjunction with a migrating filesystem,
- short-term plans for user-data replication, schema evolution, security etc [15],
- powerful data access and clustering techniques, built into the database

In addition, the company has proved to be extremely willing to work with RD45 and HEP in general and has provided excellent support The choice of Objectivity for a number of other applications, listed below, has backed up our choice of a system for prototyping

- The Sloan Digital Sky Survey at FNAL[16],
- The Motorola Iridium project[17],
- The EUCLID CAD/CAM package

8 Proposed solution to the Object Persistency Problem for LHC

The proposed solution to the problem of handling persistent objects for LHC era experiments is described below This solution, which is based upon standard, commercial components, together with a small amount of site-integration, is powerful, scalable and affordable In fact, as we shall argue, this

---

13 Which was previously known by the somewhat more amusing name of CO_2

14 O_2 does supported distributed databases, but in a non-transparent manner The user must know in which physical database the various objects or collections that (s)he wishes to access reside

15 Support for all of these features is expected during 1996 replication and schema evolution will be supported in V4, to be released in Q1 1996

16 See http://www-sdss.fnal.gov:8000/

17 See http://www.objectivity.com/releases/Motorola.html

solution is considerably cheaper than alternatives, including most particularly the development of the necessary software "in-house"

In recent months, there has been considerable interest in this solution from a number of pre-LHC experiments, such as BaBar In addition, existing experiments (NA45 and NA48) are planning to use components of the solution in production as from 1996

The proposed solution consists of

- an Object Database Management System (ODBMS) that conforms to the Object Database Management Group standard Our current preferred database for prototyping activities is Objectivity/DB, due to the unrivalled list of features that it offers, including replication of user-data, schema evolution, object versioning and support for heterogeneous, distributed environments The use of a standards-conforming product guarantees source-code compatibility between conforming products from different vendors - an application built to run with the product from one vendor may be "ported" to that of another vendor simply by recompiling (to get the appropriate header files) and relinking This capability will be demonstrated at a future Object World
- a standards-compliant Mass Storage System (X/Open or IEEE standards) Our currently preferred long-term mass storage system is HPSS However, for prototyping purposes, we plan to use both ADSM (CERN) and OSM (DESY)
- site integration, including the handling of disk caches for active data and the provision of HEP base classes

This solution offers fully transparent access to any object, or its components, in the entire, distributed, object store This includes transparent integration of the mass storage system - users need not be concerned with details regarding the physical location of the data that they are accessing, whether it is stored on disk, on tape, locally or remotely

Not only is this solution more powerful than alternatives that have been considered - none of which are capable of satisfying even the preliminary requirements listed below (see section 6), but is also more affordable

By insisting on standard interfaces, we are able to provide at least some level of independence from individual vendors By actively participating in the appropriate standards groups - and with the vendors themselves - we can influence the functionality of the products that will be available

It has been traditionally maintained within HEP [15] that commercial database systems do not, and will not, scale sufficiently to handle the vast volumes of data that will be acquired at the LHC However, database analysts and the vendors themselves consistently predict databases in the TB to PB range by the year 2000 Moreover, in the solution that we are proposing, this problem is circumvented by limiting the physical database size to some 100 GB - databases of this size are in production use today - and by using multiple, physical databases that appear to the user as a single, logical database This technique, implemented today in commercial products such as Objectivity/DB[18], provides transparent access to any object in any of the physical databases that make up the so-called *federated data-*

18 In the current version of Objectivity/DB, a federated database may contain up to 64K physical databases If each of these databases is limited to 100 GB, then the maximum size of the federation is 64 PB We expect this limit to be raised considerably in a future version of Objectivity/DB

base[19] Even transparent, cross-database references are permitted and an interface to mass storage systems is handled automatically at database open time This technique of scaling to extremely large databases is widely accepted in the database industry [23] [24] We propose, however, to limit the physical database size to 100 GB for the following reasons

- 100 GB is a very conservative limit - we can be confident that databases of such a size will be in wide-scale production use long before LHC and hence well supported by the industry,
- Filesystems will easily be able to support files of such a size (databases products typically use the native filesystem to manage the physical databases),
- Such a limit fits well with conservative extrapolations of the capacity of tertiary media - we can be confident that such media will exist and will allow us to avoid splitting physical databases across volumes,
- Given the predicted I/O rates, this figure represents a convenient "chunk" for moving to/from tertiary storage and over networks,
- The total number of physical databases required remains reasonable - some 10^5 such databases will be required per LHC experiment per year

In reality, we expect that a higher limit will be used - perhaps 1 TB or more

8 1 The Financial Model

When proposing a solution based upon commercial offerings, it is essential to consider the financial implications In particular

- Is the proposed solution affordable?
- How does it compare to alternative solutions, and "roll-your-own" ones in particular?
- How can we protect ourselves against possible problems, such as insolvency of the supplier in the future?

The financial model that we have assumed is as follows -

- The cost at "data-centres" should be "reasonable", when compared to other elements of the computing infrastructure (file-servers, tape libraries etc),
- The cost to the end-user should be no more than for other end-user software, such as compilers, run-time access to commercial libraries etc ,
- A solution where the cost for end-users is covered "centrally" is strongly preferred,
- Of the order of 50 concurrent development licenses and 1000 concurrent runtime licenses should be sufficient for LHC development and production phases,
- Solutions which require a license server and/or keys are unacceptable

Estimates obtained from both O_2 and Objectivity suggest that 150 man-years are required to write an ODBMS (although this ignores the vast number of man-years of database research that are also required) and that 50 man-years are required to write a mass storage system - the latter almost certainly being an under-estimate for a general purpose MSS This gives a total of some 200 man-years - which clearly could not be found within the HEP community for this task It is important to empha-

19 Strictly speaking, the federated database concept is more general than this See Cattell [1] for a discussion of this concept

sise that the functionality of such a system is far in advance of what is offered by existing HEP-specific packages, and, more importantly, that this functionality is *required* by future HEP experiments

The cost per LHC experiment can be expected to be of the order of $100K[20] for the necessary database development licenses and a similar figure for the runtime licenses This is clearly extremely cheap compared to the manpower that would be required to write our own system - not withstanding the long-term maintenance issues - which are totally incompatible with the manpower envelope of CERN - that such a solution would imply For example, the ZEBRA system, which cannot be compared in functionality to that offered by Object Databases, has required more than 10 man-years in development and support costs (A more reasonable comparison might be ZEBRA+FATMEN+HEPDB+event directories, although even in this case, the functionality does not approach that offered by an ODBMS+MSS based solution)

In addition, it is our belief that the use of database features, such as user data replication and data caching, obviates the need for bulk data import/export This alone results in significant savings in hardware - the import/export stations, and, equally importantly, in the manpower that would otherwise be required to manage the import/export, including the associated book-keeping

In HEP, in-house software has often depended on a single person or small support group for maintenance and development This is clearly not a viable solution in the future, particularly in the light of the extremely long timescales involved in an LHC-era experiment A standards based commercial solution offers significant advantages in this respect

8 2 Future of ODBMS vendors

A few years ago, the long-term future of ODBMS vendors was far from assured However, this is no longer the case today - the commercial demand is now sufficiently large that the future of the technology can be assured Objectivity/DB, for example, has been chosen for a large number of important applications, including the Motorola Iridium project, the Sloan Digital Sky Survey, plus numerous applications in the Telecoms industry, control systems (e g O'Hare airport), Engineering Data Management (e g the Matrix system, under consideration for a future EDMS system at CERN), etc Moreover, the standard interface (both API and database exchange format) would permit a migration between products, access to the source-code in the event of collapse of the company can be written into any contract

8 3 Manpower issues

As explained above, a solution based on commercial products results in significant savings in manpower both at development and maintenance stages However, it will still be necessary to provide central support for the system, both on the database and mass storage sides Thus, we envisage the equivalent of the current RDBMS support section in CN division, but focusing on ODBMSes This

20 One may also compare the cost of such a solution with that of the physical media required to store 10 PB Using IBM s Magstar technology, one million such volumes would be required, at a total cost of 65MSF (1996 CERN prices) Even with an optimistically predicted hundred-fold increase in cartridge capacity at no additional cost per volume, the cost would be 650KSF!

compares favourably with the numbers required to support existing, much less functional, packages In addition, experts will be required within the collaborations to manage experiment specific details, including handling schema, replication rules, security issues and so on Depending on the computing model that is adopted, one can expect a reduction in the effort required on behalf of the experiment, particularly in the area of data import/export and in overall data management (which currently involves a small fraction of a large number of people, in addition to a few essentially dedicated personnel) The overall saving to HEP in general - in particular, the amount of time that physicists will be able to devote to physics analysis and not data management issues - can be expected to be significant, if not highly visible

9 Progress on Milestones

9 1 Requirements specification

At the time of the interim report made to the LCRB in November 1995, it was realised that the first milestone, namely to produce a 'requirements specification for the management of persistent objects typical of HEP data, together with criteria for evaluating potential implementations', could not realistically be met until late in 1996, when the Computing Technical Proposals of the ATLAS and CMS collaborations will become available However, given that the work of RD45 could provide useful input to the process of designing the computing models, the following complement was added to the first milestone

- Produce a 'Statement of Probable Capabilities' for a HEP persistent object management system based upon commercial ODBMSes and large-market mass storage systems'

9 2 Probable capabilities of a HEP object manager

As a response to the revised milestone described above, a document, entitled "Object Databases and Mass Storage Systems the Prognosis" [12], has been produced, which outlines the likely evolution of disks, filesystems, tapes, mass storage systems and object databases over the next few years In this report, we have used conservative estimates - in many cases, we believe that the actual evolution will be more rapid than we have anticipated - and we have largely limited our predictions to around the year 2000 Nevertheless, we foresee that the appropriate technology, that will allow us to build and extremely powerful HEP Persistent Object Manager, will be available several years before LHC data-taking commences Furthermore, many of the basic components that are required are already in production today! A brief summary of the key points of the report is given below

- Disks with a capacity of 100 GB or more per drive will be in production use before 2004 It will be possible to build a disk farm, comprised of multiple disk pools spread over many disk servers, with a total capacity of several hundred TB,
- Tapes with a capacity of 100 GB to 1 TB, with a throughput of 50-100 MB/second, will be available,
- Parallel filesystems, with single stream throughputs in excess of 100 MB/second will be available,
- At least one mass storage solution, probably HPSS, will provide the capability to scale to the PB region, with data access rates in the hundreds of MB/second range,

- A number of Object Databases, conforming to the Object Database Management Group standard for the underlying Object Model, C++, ODL and OQL bindings will be available (In fact, 10 database vendors have announced at least partial ODMG compliance for 1996) Many powerful capabilities, such as user-data replication and schema evolution and object instance migration, together with the ability to scale to the PB region (by using distributed database technology and the federated database concept in particular) will exist,
- A number of mechanisms will exist to interface Object Databases to Mass Storage Systems in a fully transparent manner

9 3 Evaluation of the ODMG's ODL

The second milestone established by the LCRB for RD45 called for an evaluation of the Object Database Management Group's (ODMG) Object Definition Language (ODL) In order to perform such an evaluation, we attempted to describe a number of existing data models using this language In virtually all cases, these models were not designed using an object-oriented approach However, pending a true object model for a HEP event, we were constrained to use these models It is our conclusion that the ODL is indeed both suitable and sufficient for describing the object model of HEP events

The Object Definition Language (ODL) is a language used to define the interfaces to object types that conform to the ODMG Object Model ODL is not intended to be a full programming language, but rather is a specification language for interface signatures It is in fact a DDL (Data Definition Language) for object database schemas It defines the characteristics of types, including their properties and operations ODL is intended to define object types that can be implemented in a variety of programming languages For example the ODL bindings to C++ and Smalltalk are designed to fit into the declarative syntax of those programming languages

ODL is based upon the Interface Definition Language (IDL) defined by the Object Management Group (OMG), for use with Object Request Brokers IDL itself is based upon C++ syntax, and thus ODL also retains a C++ flavour However, ODL files should not be confused with C++ header files[21]

- ODL is not tied to a particular programming language ODL can be used to describe objects that are created/manipulated from a variety of languages, today for C++ and Smalltalk, with possible future bindings for Java and SQL3,
- ODL extends IDL It adds the constructs required for specify the complete semantics of the ODMG Object Model,
- Additional keywords include *extent, keys* and *relationship* (*Inverse* is also defined, but only has a meaning when used with the *relationship* keyword)

Tools already exist[22] which permit the automatic generation of ODL files from an object model - the model itself contains sufficient information to generate not only plain C++ header files, but also ODL files Future tools, such as those which support the "Unified Method" - the recently announced meth-

21 In trivial cases, the definition of a class in ODL and the equivalent C++ header file may, in fact, be essentially the same The minimum change to a standard C++ header file to permit the creation of persistent instances of a give class is to inherit from *d_object*

22 e g Classify/Views from Micram Technology, the distributors of Objectivity/DB in Germany

odology, still under development, that combines the ideas of the Rumbaugh and Booch and other leading methods, can also be expected to support ODL generation

Unlike, for example, the ADAMO DDL, ODL does not support validity ranges for data members This does not imply that validity ranges cannot be supported by an application that uses an ODBMS for persistence - this can easily be accomplished, e g using accessor functions In addition, support for validity ranges is currently being discussed within the ODMG in the context of an Object Constraint Definition language

In order to test the suitability of ODL for describing an object model of a HEP event, we used existing data models from ATLAS, DELPHI, H1, L3 and OPAL We note that different data structure managers and data definition languages were used by these experiments, and so we were also able to compare ODL against several existing DDLs To convert the existing data models[23], a program has been written, which automatically generates Objectivity/DB DDL[24] files starting from the data definitions of existing events As a separate step, described below, we then loaded existing data into an Objectivity/DB database

Even though the data design of experiments differs significantly (e g in the number of bank types, use of links between banks, use of replicated and variable length substructures, strong link typing) it was possible to express all data models in Objectivity/DB DDL

In summary[25], we were able to successfully describe the event model of all of the above experiments using ODL However, as these models were not developed according to the OO paradigm, we used only a small part of the full power of ODL

9 4 Prototyping activities using object models described using ODL

In order to test the suitability of a commercial ODBMS for storing HEP event data, we built a number of prototypes, both using existing event data (H1, L3, OPAL), and also various models based on the ATLAS electromagnetic calorimeter Unfortunately, we do not yet have a true object model for a HEP event (although such a model is currently being developed as part of the NA45 experiment), and were thus constrained to base our work on existing event models

9 5 Prototypes using H1 and OPAL event data and Objectivity/DB

As described above, (see section 9 3), we converted the event descriptions from a number of existing experiments into ODL Starting from these descriptions, we then loaded existing data into Objectivity/DB To perform this work, two interfaces between the existing, Fortran-based, data management systems were required

---

23 Currently, both the H1 and ZEBRA (DZDOC) DDL formats are supported

24 This DDL is very similar to the ODMG ODL, which is currently not directly supported by any ODBMS The current version of Objectivity/DB supports ODL via a product from its German distributor A future version will support ODL directly

25 A more complete description of this work carried out by Witek Wajda/Krakow, is available in /afs/cern ch/rd45/evaluations/odl_eval tex

- routines to read existing data and convert them to C++ classes, created according to the ODL definitions,
- routines to read data from the database, and re-create the appropriate data structures in memory (needed only to permit existing Fortran applications to run againsts the database and provide a consistency check on the conversion process)

Combined, these two interfaces allowed us to functionally replace the ZEBRA FZ package - by linking with the appropriate library, an existing application could move to using an Object Database for persistence without any source code changes It is important to stress that we are not proposing to use this prototype for production - it was developed purely as a means of verifying the event structures that we had defined in ODL As such, and pending a true object model of a HEP event, this work was addressed at fulfilling milestone 3 (see section 1 and section 9 4) These prototypes used both FPACK, developed at DESY (for H1 data) and ZEBRA (for all other data)

The generated class description in Objectivity/DB DDL contains data members with types, names and relations to other classes as deduced from the documentation The program has been used to convert data structures from a number of existing experiments, including H1 and OPAL corresponding to 790 and 973 bank types respectively In both cases, there were some 6000 named data members in the corresponding database schema

Using this conversion technique we have produced object databases containing up to about 1 GB of experimental data[26] The size of the databases created indicates an overhead of some 5% over the volume of data stored

9 6 A Prototype using L3 data and Objectivity/DB

In order to test the use of the Objectivity/DB ODBMS as an event store, part of one of the L3 data structures has been modelled into Objectivity/DB schemas A small prototype has been built to write L3 events into an Objectivity/DB database and to read them back The purpose of this prototype was twofold

- to test the use of Objectivity/DB's DDL (ODMG's ODL) to describe HEP data structure,
- to test the performance of a commercial ODBMS as event store

For these purposes, we chose to model each "reconstructed object" as a persistent object The original relationships between the various entities (runs, events, reconstructed objects, hits) have been modelled using Objectivity/DB object-associations Each object has been split into three parts according to the frequency of access in typical analysis programs Single objects are thus quite small, ranging between three and twenty words in size Objects belonging to different classes have been stored in different containers which, in turn, have been arranged into three different database files according to the presumed access frequency

We have found that

- Objectivity/DB's DDL is perfectly adequate to describe the logical structure of HEP (L3) events,

26 The limiting factor for the database size was the hard disk in our setup

- A single Federated Database, composed of several tens of database files, can be navigated transparently without any knowledge, at reading time, of its physical structure it is sufficient to open a single object and follow the associations object to object till one reaches the desired information,
- Objectivity/DB is a robust software system documentation is very clear, several tools are provided to manage the database and the user application actually never crashed

On the other hand, there are still some open questions related to performance and the physical structure of the databases For example, with the data model used, reading only a subset of the events (one out of ten, for example) did not produce a comparable timing improvement and calls for further investigation However, we were able to confirm that the use of a fine grain object structure does indeed produce the expected time gain if one reads only a part of the event structure

Future investigations will include the implementation of "transient" object attributes, the study of various query mechanisms and "data-mining" techniques

9 7 Evaluation of different persistent object models with Objectivity/DB

In order to direct the design of the ATLAS offline reconstruction software, we conducted a series of tests with Objectivity/DB The purpose of these experiments was to determine an appropriate granularity for persistent objects, to understand what kind of effects different database and run-time configurations have, and to get some sort of a picture of how tightly we should couple our applications to the database We were mainly interested in relative costs between different designs, since absolute speed values have little meaning in the current design stage However, we did use the test results to set the performance scale we want to target at

All tests were run on a HP 712/80 with 64 MB of memory The test program was the only active process on the machine

We experimented with three different object models in eight different database configurations, and each test was repeated 10x2x2 times One model was used to determine the maximum attainable performance Another model made every calorimeter cell a persistent object The last model was a compromise between the two, using a flexible, persistent container to hold transient cells[27] The size of the data structures was that of the electromagnetic calorimeter (~200k cells), and they were tested with typical operations used in the reconstruction process

Our conclusions are

- Making the smallest detector elements, such as cells in the calorimeter, persistent causes at least an order of magnitude performance hit If the objects are also small, for instance a single floating point value, the space overhead becomes unacceptable
- Container objects are able to combine flexible abstractions and reasonable speed

27 That is, the detector cells themselves had no persistence capability They only achieve this by being part of another persistent object, which in this case is the container

- At lowest application levels it might be a good choice to separate database representation of data from that used in algorithms The separation must be kept limited to low-level objects in order to get the maximum benefits of having an object database

More related to C++ compilers than to Objectivity/DB was the verification of the assumption that it is the initial abstraction that is most expensive Further costs of using the abstractions -- such as STL-like iterators -- tend to be a lot less significant Currently, it is not the ODBMS technology but the compiler technology that is limiting the use of interesting abstractions

9 8 Performance measurements using existing data and Objectivity/DB

When considering the performance measurements listed below, it should be noted that this simplistic conversion scheme, particularly the automatically generated object model, provides by no means optimal database performance Important design parameters, such as the database page size, the mean object size, the implementation of variable length objects, the number and implementation of object relations are taken from the bank documentation or are chosen ad hoc for all classes

The database performance estimated using this model is therefore a lower bound for the expected performance of a purpose built and optimised object model We also expect a significant gain from a careful study of the effect of various object models on database performance

All measurements have been performed on a HP 712/60 Workstation with 64 MB main memory and a locally connected hard disk containing input event data and the database files

9 9 Sequential read performance

The sequential transfer performance of the disk and file system were estimated by reading a 200 MB file in stand-alone mode as ~1 5 MB/sec The performance measurements have been normalised to this rate

The read performance of conventional I/O packages (ZEBRA and FPACK[28]) was determined by measuring the time spent to read all events from the dataset sequentially

The read performance of the database was measured using a scan through all database objects, which corresponds to accessing every bank in each event in a conventional data structure manager

As table 1 shows, the overall read performance of the database is comparable with conventional packages during the initial read and even better during following reads, due to the effect of the cache[29]

9 10 Performance Scaling with Database Size

The scaling of read access time with database size is an important issue As is shown in figure 1, access time to objects in an Objectivity/DB database is essentially constant up to 300K objects -

28 The measurement for the FPACK package does not include any memory management

29 The gain by caching depends heavily on cache size, total size of the database and amount of data from the object

approximately the number of event objects that can be expected in a 100 GB physical database (For small databases, the effects of the cache dominate, which explains the initial slope)

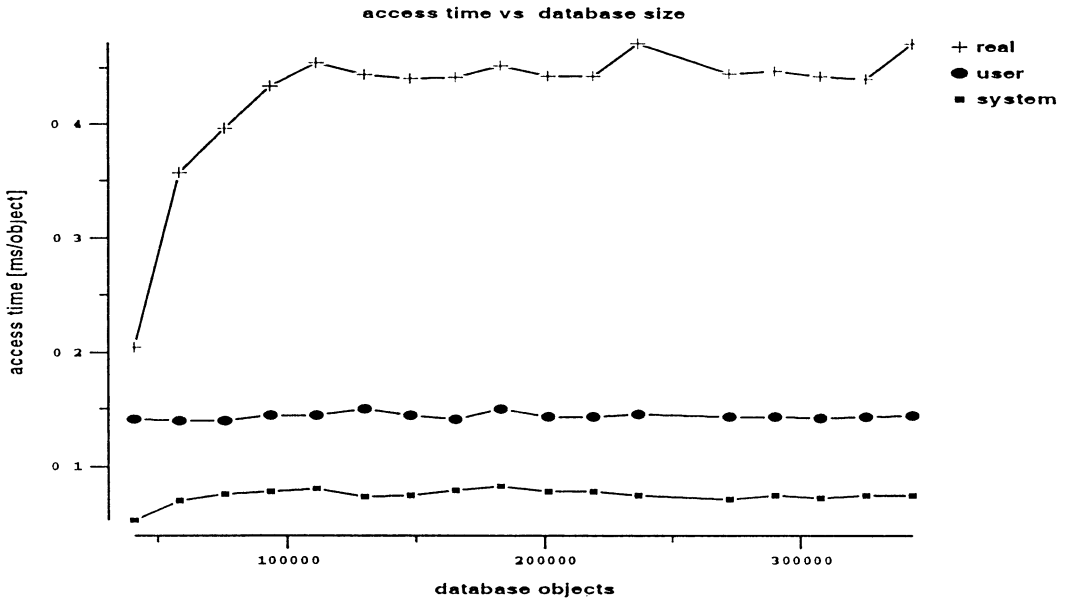


Fig 1, variation of access time with database size

9 11 Write performance

A direct comparison of the database write performance with I/O systems like FPACK and ZEBRA is not meaningful since the functionality of these systems is very different

- Sequentially writing events to disk using a conventional I/O system does not involve much book keeping A large fraction of the time spent is consumed by file allocation and raw transfer inside the operating system On the other hand, the sequential nature of traditional packages does not allow events to be deleted nor for new information to be added to existing events[30],
- A database supports concurrent writing of several processes, e g in the case of processors in a filter farm Neither of the I/O packages mentioned above, nor existing operating systems provide transaction based consistency control that is provided by a database

To simulate the effect of the "commit", one can issue a "sync" system call after each event, which ensures that all unwritten buffers are flushed to disk Issuing a sync after writing each event with a package such as FPACK or ZEBRA results in a significant loss in performance Similarly, issuing a commit after a larger number of events, rather than after each individual event, gives markedly improved performance in the case of Objectivity/DB However, more significant effects can be obtained from a variation in the data model - models optimised for write tend to show poor read performance, and vice-versa (A model optimised for write performance gives essentially the same I/O rate as raw I/O to the disk) In our case, it is clearly more important to optimise for read access, although write performance cannot be ignored entirely Although we have performed a number of

30 The current data re-processing strategies of most experiments therefore involve copying nearly all data even though only a small fraction of data is actually updated

measurements with different models, we do not feel confident in presenting the results obtained at this stage, but plan to continue with detailed studies in the immediate future

Table 1: read access performance

IO rate [kB/sec]	read access
disk & fs	1560
FPACK	1146
Objectivity	1250

9 12 Performance measurements of C++ and SQL++ API's to Objectivity/DB

[The following information was extracted from a report performed as part of the SDSS project at Fermilab We include it here, with the permission of SDSS, as we feel that it is directly relevant to RD45's activities]

The relative performance of C++ and SQL++ API's for accessing Objectivity/DB databases was measured It was hoped that the SQL++ API would offer similar performance to C++ programs performing comparable functions Unfortunately, the SQL++ API performed about ten times as badly as the equivalent C++ interface An example of such a query, executed against a 74MB database containing 100,000 objects of a single class, making a range selection that returned 1524 objects, showed the following (all times system+user)

- C++ API using predicate query 1 00
- C++ API, range query performed in code 0 57
- SQL++API 10 48
- "dump" of DB file to null device 0 57

This information has been fed back to Objectivity, who have acknowledged receipt and will investigate As the figures stand, they suggest that the SQL++ API cannot currently be used for performance reasons However, it is believed that SQL++ statements can be translated into the equivalent C++ API calls at run time, offering an alternative solution

9 13 A prototype of an event-tag database

Using the databases described above, a number of scenarios for data access have been prototyped The prototype itself corresponds to less than 3000 lines of C++ Code and DDL

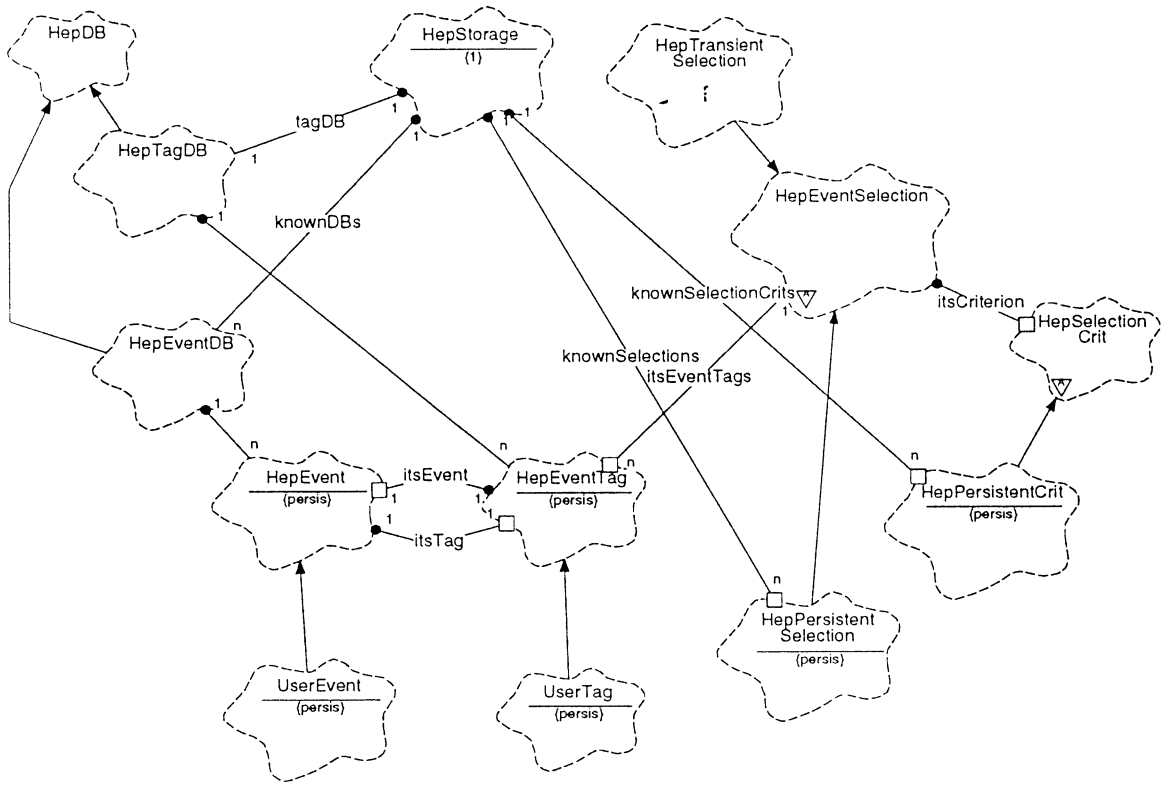
These scenarios include

- selections based upon tag objects, where the selection returns a collection of event objects, which in turn can be used to access the event data,
- deep selections, where the selection is based on the actual data itself

In both cases, the collections may be either transient or persistent Persistent collections may be accessed, for example, by name, e g "TwoPhotonEvents" In addition, the criteria used to established the collections are objects in their own right and can themselves be transient or persistent A further refinement of this prototype, which is expected to have important performance implications, would be to (optionally) copy the data referenced in the selection process, re-clustering on the fly for subsequent access

Further work on this prototype, including detailed performance measurements, is planned for the future However, to obtain realistic results, we feel that it is essential to start with a realistic object model of a HEP event, rather than a simple conversion of an existing data structure

The class diagram corresponding to this prototype is shown below



9 14 Conclusions

Our conclusions from these tests are as follows

- The size of the database created in this way[31] is only a few percent larger than for existing, HEP-specific solutions,
- Sequential read access to data stored in an Object Database is at least as good as with existing, HEP-specific systems (usually slightly better),
- Access to individual objects, or attributes of individual objects, is possible with little overhead - as was expected, the required I/O is the dominating factor,
- Database functionality allows us to implement systems that are significantly more powerful and consistent than today's solutions, including the implementation of functionality similar to that offered by e g event directories, but in a manner that fits naturally within the programming environment
- Detailed questions still remain, largely related to database structure and its impact on performance, and these require further study and prototyping

10 Standards Activities

The following standards were identified as relevant to RD45

- C++,
- The Object Management Group (OMG),
- The Object Database Management Group (ODMG),
- IEEE Storage System Standards Working Group (SSSWG)

No attempt was made to join the C++ standardisation effort - it being beyond the scope of RD45, although copies of the draft standard have been obtained and studied

CERN joined the OMG as an associate member, which gives us the right to attend all meetings (although none have actually been attended so far, primarily because they normally take place in the US), and we obtained copies of all OMG documentation

CERN also joined the ODMG as a reviewer member[32], and we have attended all full meetings since July 1995, as it was our conclusion that the work of the ODMG was very closely related to our activities and that it was important to both understand and influence the future direction of this standard

Involvement in IEEE Storage activities by members of RD45 pre-dates the project by many years These activities have included the organisation of the first (and only) SSSWG meeting in Europe (held at CERN during 1993), participating in the organising committees for the past 3 symposia, the

31 Which, being based on a simple conversion of an existing data model, is clearly not optimal

32 Full membership is only available to vendors who ship an ODBMS and who undertake to conform to the ODMG standard in a future release

local organiser of the 1st International (13th IEEE) Symposium on Mass Storage Systems (held in Annecy during June 1994)[33]

For further details regarding the status of these various standards, the reader is referred to the report "Object Databases and Mass Storage Systems - the Prognosis", produced in response to the revised milestone 1 (see section 3 3)

11 Technical presentations and Training

Following the identification of the appropriate technology, a number of technical presentations and training courses, the latter primarily oriented towards RD45, were organised at CERN These included

- Presentations on the O_2, Objectivity/DB and ObjectStore ODBMSes,
- Presentations on the Object Request Brokers (ORBs) from Iona (ORBIX) and DEC (ObjectBroker),
- A 5-day course on Orbix[34] Places were made available for one representative from both GEANT-4 and CICERO projects,
- A 3-day course on the O_2 ODBMS,
- A 3-day course on the Objectivity/DB ODBMS for C++ programmers This course was repeated for ATLAS (largely, but not exclusively ATLAS-MOOSE), and was also held, in "hands-off" form, in the CN auditorium and was attended by some 30 people

In addition, the standard OO training schedule, given by John Deacon, was followed, as well as training on the ESA Software Engineering Standards[35]

These presentations and training courses enabled us to understand the major differences between the various products - typically related to architectural design issues in the case of the various ODBMSes - and of the applicability of OMG and ODMG standards to the problem of persistence

A 5 day workshop, focusing on prototype applications with Objectivity/DB, was held during the February 1996 The workshop programme can be accessed via the URL http //wwwcn cern ch/pl/cernlib/rd45/workshop/objy_feb96 html

12 Work on LHC computing models

RD45 is represented in the computing model working groups of both ATLAS and CMS Links have also been established with both ALICE and LHC-B, so that the appropriate information exchange can also occur with these collaborations A number of presentations have been made to both ATLAS and CMS[36], and indeed the main purpose behind the revised milestone 1 (see section 3 3), which has been

33 Future plans call for the 3rd International Symposium, which is to focus on the integration of databases with mass storage systems and high performance data management issues to be held in Annecy in the second quarter of 1997

34 A CORBA-compliant Object Request Broker from Iona Technologies, Dublin

35 The material presented in this course is largely covered in the books "Software Engineering Standards" [20] and "Software Engineering Guides" [21]

distributed for comments and corrections to both groups, was to provide information exchange regarding the various technologies being studied within RD45

It is clear that very close work with these groups will be needed in the future

- to help in the generation of the computing technical proposals,
- to help prepare user and system requirements regarding persistent object management at all stages of production (reconstruction, alignment, calibration, analysis etc)

We feel that it is our role to understand how the various scenarios discussed in the Computing Model Working Groups can be implemented in an Object Database environment, and to feed-back technical information regarding Object Database capabilities, such as replication, schema evolution, object versioning and so forth, to these groups

In addition to progress reports and the overall model being developed within RD45, the main topics presented to the ATLAS and CMS Computing Model Working Groups were

- a strategy for building a multi-PB HEP object store, that offers transparent, object level access, using federated databases together with mass storage systems,
- distributed databases, and user data replication

13 Mass Storage related activities

13 1 Project-X

A study of a pre-release of an IBM development, known as Project-X, was carried out This software sits in front of a standard Mass Storage System - IBM's own ADSM in the pre-release, and provides meta-data driven access to data The meta-data is stored in a front-end database - DB2/6000 in the prototype, although there were plans to support other relational and even object databases for this task The functionality of this package can be compared to that of the FATMEN package, developed at CERN

The first tests with this software showed up a number of weaknesses, both in the actual implementation and in the design itself A report detailing our findings was sent back to IBM, and a second version, which claimed to address many of the issues involved, was received For a number of reasons, the new version was never installed However, at approximately the same time, internal IBM reorganisation caused the project to be cancelled

Nevertheless, this exercise taught us two valuable lessons

- the importance of not being locked in to any one vendor or product and the importance of working with standards-compliant products This is, of course, nothing new, but this experience provided a very strong reminder!

36 See http://wwwcn.cern.ch/pl/cernlib/rd45/slides/index.html

- the dangers of confusing the functionality required in the different components To a very large degree, this project attempts to add database-like functionally to a mass storage system Our conclusion from this and from numerous discussions at the recent IEEE symposium on Mass Storage Systems[37], are that one should restrict database functionality to the database layer, and minimise one's requirements on the mass storage system This results in greatly simplified, and thus more achievable, requirements on the mass storage system (see section 6 3)

13 2 IEEE related activities

Some of the key requirements of LHC experiments in terms of mass storage were presented in a paper[38] given at the 14th IEEE Symposium on Mass Storage Systems in Monterey [13] This paper underlined some of the key problems with the commercial response to the mass storage requirements of large sites, and problems with the standardisation effort itself

- There is insufficient commercial demand to justify the necessary investment to produce a high-end mass storage system (capable of handling the requirements that are being addressed by e g HPSS) - it is therefore unwise to assume that a commercial solution to the mass storage problem will exist,
- Standards for meta-data are not required - these should be handled by the database, just like any other data,
- A mass storage system that is build according to the IEEE reference model addresses only a small part of the total data management problem Site integration and disk pool management, the latter being particularly important in our environment, must still be provided,
- Standards are urgently required for exposed interfaces, and not internal interfaces (where the effort has been concentrated)[39],
- A combination of Object Database technology together with mass storage technology *and* appropriate site integration is, nevertheless, sufficient to satisfy the data management needs of LHC era experiments

13 3 The HEPMSS group

The HEPMSS group was established during a "birds-of-a-feather" sessions at CHEP '95 Its goal is to facilitate the exchange of information concerning mass storage systems and their deployments within the HEP community A HEPMSS web page has been setup[40], and a mailing list established It was also intended to hold a mini-workshop at SLAC, coinciding with an ODMG meeting in Monterey, to discuss the mass storage requirements of the various laboratories and to attempt to define common requirements[41] A further topic for this workshop was to increase HEP involvement in the IEEE standardisation effort It is highly regrettable that this workshop had to be cancelled, due to lack of feedback Given the importance of this area for HEP, it is the recommendation of the RD45 collabo-

37 The integration of databases with mass storage was a hot topic at the 14th IEEE symposium, October 1995

38 The presentation can be viewed via the URL http://wwwcn.cern.ch/pl/cernlib/rd45/slides/ieee95.ps

39 In this context, we note the X/Open draft XBSA API [22] and the de-facto IBM/STK tape drive/silo interfaces

40 See http://wwwcn.cern.ch/~hepmss

41 The draft CERN requirements can be accessed through the HEPMSS page - http://wwwcn.cern.ch/~hepmss

ration that further steps be taken to improve collaboration between the laboratories on mass storage issues and that participation in IEEE conferences and the standardisation working groups be increased

14 Future Activities

Many of the activities described below are experiment independent However, it is clear that prototyping and other investigations in these key areas is required to help the experiments define and refine their computing models We, therefore, describe first the experiment-independent activities, followed by experiment-specific plans

14 1 Mass Storage Interface

A transparent interface between the database and mass storage system is key to the use of database technology for very high volumes of data The preferred solution, namely using database faulting, whereby the database open failure is trapped, and site-specific code to *stagein* or recall the appropriate database is issued in the handler This technique retains full transparency, even in the case of cross (physical-)database references A number of interfaces will be evaluated, including the use of the existing CERN staging software, as well as the archive component of IBM's ADSM product

In addition, DESY plan to test Objectivity/DB with their production mass storage system, based upon OSM

14 2 3rd International IEEE Mass Storage Symposium

It is planned to hold the 3rd International IEEE Symposium on Mass Storage Systems in Annecy, shortly after CHEP '97 (i e May/June 1997) The theme of this conference will be the integration of database management and mass storage systems Although the orientation of the conference cannot be purely HEP, it is clear that this meeting, albeit too late for the Computing Technical Proposals of ATLAS and CMS, could be of significant help in understanding the technology, the status of standards, and future directions

14 3 Performance measurements with parallel filesystems

In order to understand how parallel filesystems impact the performance of an ODBMS, we intend to measure the performance of Objectivity/DB using the parallel filesystems on the CERN CS-2 and later SP-2 This will be compared with the performance obtained on sequential filesystems and against the raw I/O rate from the parallel filesystems themselves The affect of using different block sizes, both at the database and filesystem level will also be investigated

14 4 Tests of replication

Replication, both of user and system data, is expected to be a key feature in the successful deployment of databases in a distributed environment The basic features of replication, including perform-

ance, will be tested both in the LAN and WAN The robustness of the implementation to network failure will be studied, as will update access to replicated objects

14 5 Tests of schema evolution and object instance migration

Given the long time period over which the LHC experiments will exist, it is highly likely that the schema will change Databases normally provide built in mechanisms for detailing with schema evolution and of the migration of objects created according to these schema We intend to perform a detailed study of the Objectivity/DB implementation of schema evolution, add to compare it against the requirements of the LHC experiments

14 6 Object Versioning

Another database capability that is potentially interesting, particularly for handling calibration data, is that of object versioning We intend to investigate the support for object versioning in Objectivity/DB in the context of a calibration database As there are a number of experiments expressing the need for such a database, we also intend to produce a common set of requirements, starting from those produced for BaBar[42]

14 7 Data organisation and access techniques

An area that is known to be critical for performance is that of data organisation - different object models and clustering can result in widely different performance characteristics, and it will be extremely important to master the various techniques involved Similarly, the data access mechanisms that are provided by the database (indices, naming etc) will be studied so that optimal performance can be achieved

14 8 Investigations of database parallelism

For performance reasons, parallelism will be used in many areas - event reconstruction will be performed in parallel, databases will exploit parallel I/O (parallel filesystems), transfers to/from mass storage will be done in parallel, as will network transfers In addition, the questions of parallel database update and query will be fundamental It is intended to study these areas, in the context of an event reconstruction farm and data analysis respectively

14 9 Data mining

Data mining[43] is a technology that is being increasingly important in the commercial world Data mining is the efficient discovery of previously unknown relationships and global patterns in large databases This information is then used to make business decisions

Data mining techniques include the following

42 See /afs/cern.ch/rd45/babar/calibdb.txt

43 See for example, http://www.almaden.ibm.com/cs/quest/index.html

- associations,
- classification,
- sequential patterns,
- similar sequences

The term "data mining" has been applied to the process of scanning through large volumes of physics data, selecting events with certain characteristics and looking for patterns *within* individual events This is not data mining in the normal sense Nevertheless, it will be important to understand whether techniques developed for the commercial world - which will clearly drive the market and directly influence future database technologies - can be exploited in High Energy Physics

14 10 Data analysis and visualisation

An important step in data analysis is visualisation We will investigate how data residing in the database can be efficiently queried and past into the visualisation framework, such as Iris Explorer from SGI, which is based upon the de-facto graphics standards OpenGL and OpenInventor

14 11 GEANT-4

The GEANT-4 (RD44) collaboration has been approved for a second year, with the following milestone

- Release by early 1997 a first version of Geant4, with tracking, geometry, EM processes at least equivalent to Geant 3 for simulation of events in LHC detectors

Although not a milestone on the RD44 project, the following statement was added as a comment by the referees

- The first version should have I/O as provided by RD45

This implies close collaboration with RD44 during 1996, and the installation of appropriate database software and licenses at a number of remote sites, including LBL, KEK and Manchester University in the UK

14 12 Production use of Objectivity/DB as input to analysis (NA45)

The NA45 collaboration has recently taken the decision to move to C++ A production run is planned for spring 1996, where a few TB of data will be reduced to some 200 GB of data The production will be performed on the CERN CS-2, and it is proposed that the output data reside in Objectivity/DB databases These databases, which will be limited to about 1 GB in size, will be manually staged to tape for export to outside laboratories where they will be used as input to physics analysis Although the volume of data concerned is relatively small, this will allow us to test the use of object databases in a production environment, and as input to the analysis stage In addition, as all physics data is stored on parallel filesystems on the CS-2, it will allow us to investigate the use of an ODBMS with

such filesystems Finally, although not required for the initial production run, it will allow us to prototype the mass storage interface

14 13 Production use of Objectivity/DB for a calibration database (NA48)

NA48 are interested in the possibility of using Objectivity/DB to store calorimeter calibration data on their online system Although the existing requirement is to store C structs (Objectivity/DB also provides a C API), longer term plans call for a C++ interface Although the data volume is again very modest (at the level of MB or tens of MB), it will provide us with

- an additional production deployment of ODBMS technology - this time in an online environment,
- early production deployment of the proposed calibration database

Other experiments have also expressed interest in a calibration database, including ATLAS, BaBar and CMS An attempt to define a common set of requirements and, hopefully, to provide a common solution will also be made

14 14 Storing of "legacy data" in Objectivity/DB

When an experiment in HEP stops taking data, members of the collaboration move over to new experiments - the programs are no longer maintained, and sooner or later, the data become unusable and eventually the physical storage media on which they reside are destroyed or are no longer readable The Crystal Barrel collaboration propose to retain approximately 1 TB of 'good' DST information, plus the programs that process the data, essentially for ever Although these data will be processed only rarely, if ever (apart from tests), this project will allow us to test the mass storage interface that will eventually be required

The current proposal is as follows

- convert existing data definition into Objectivity/DB DDL,
- define schema in Objectivity/DB database,
- use "loader" to convert data from ZEBRA FZ format to Objectivity/DB,
- archive individual Objectivity/DB databases using IBM's ADSM product,
- create a meta-data database which will allow individual Objectivity/DB databases to be retrieved from the archive as required,
- data access will be provided by the "FZ-compatibility" layer on top of an Objectivity/DB database The application code will remain unchanged, and call routines such as FZIN However, by linking with a different library, the required data will be transparently loaded from the database and converted into in-memory ZEBRA data-structures

Although this does not offer the same level of functionality as if a tag database were used (see section 9 13), it matches the requirements of Crystal Barrel The current programs are based upon file-level access, and they have no plans to rewrite their code to benefit from object-level access

14 15 ALICE

The ALICE collaboration is currently represented in RD45 by one part-time member, who is working on a project to store ALICE test beam data in an object database

ALICE test beam data have been taken with the WA98 data acquisition system, and a library of C++ classes and routines to work with the WA98 raw data format have been readily available Although this test is not a commitment by the ALICE collaboration to the WA98 data format for the mid-term future, about two hundred Gigabytes of test beam data in this format exist

The data volumes expected from the ALICE detector systems (10^{15}-10^{16} bytes) are much higher than in today's typical applications, and also higher than those of other LHC experiments The data volumes involved are expected to be quite manageable by the time the LHC experiments start taking data Currently, the tests with the Objectivity/DB database serve three main purposes for the ALICE collaboration

- to prove that an ODBMS works in a reliable fashion,
- to help make the ALICE collaborators, many of whom still use Fortran/C as programming languages, acquainted with object-oriented programming techniques,
- to guide the collaboration in the selection of a database package which stores the properties of the ALICE detectors (geometry, materials,) It is foreseen that this database will be used for simulations and also during the analysis of the detector test data, and will be a candidate for the database standard within the collaboration for the mid-term future

14 16 ATLAS

ATLAS have expressed strong interest in RD45's research into schema evolution, object instance migration, object versioning etc and it is anticipated that the results of these investigations will help to develop further the Atlas computing model

In addition, work on the Computing Technical Proposal is expected to be a major item of work during 1996

14 17 CMS

In 1996 CMS plans two major activities related to persistency

- Use Objectivity/DB to store "hits" output of CMSIM (CMS Fortran simulation program) and use them as input to CMS RD41 reconstruction prototype, The use of Objectivity/DB to store higher level reconstructed objects (tracks) is also foreseen
- Test Objectivity/DB as a "Calibration Database" This database can also be used to store other event-independent, (but time and/or version dependent) objects like geometry items, configuration parameters etc

14 18 LHC-B

Activities with LHC-B are expected to commence during 1996 To this end, a presentation to the collaboration on the work of RD45 is being planned and two members of the LHC-B have been added to the RD45 mailing list

15 Proposed Milestones for 1996

In addition to the requirements specification - carried over from the first year, the following milestones are proposed for the second year of the RD45 project

- Demonstrate the use of an ODBMS in a production environment as input to the analysis stage of an existing experiment (NA45),
- Develop a mass storage interface capable of satisfying the requirements of Crystal Barrel, namely to store around 1 TB of data in multiple object databases, with transparent access to tape resident databases,
- Coordinate the effort to produce a common set of requirements for a calibration database and produce one or more prototypes based upon these requirements,
- Provide a persistence service to the RD44 (GEANT-4) collaboration,
- Investigate the use of object databases with parallel filesystems, object versioning and schema evolution capabilities,
- Provide appropriate input to Computing Model Working Groups to assist in the development of the Computing Technical Proposals

16 Conclusions

We have identified a powerful, scalable and affordable solution to the problem of managing persistent objects for LHC-era experiments This solution, which is based upon the combination of standards-conforming commercial Object Databases and Mass Storage Systems, has now been widely presented within the HEP community

Starting with physics data from NA45 in 1996, and continuing with calibration data from NA48 and perhaps also ATLAS and CMS in 1997 and beyond, there are plans to deploy elements of the proposed solution in production in a number of HEP experiments between now and LHC We can expect to learn much from these early deployments of the technology within HEP, which will also help generate confidence in the validity of the proposed solution

However, there are still many questions that need to be answered, primarily related to performance, and we intend to focus on these issues during the coming year We are confident that we can gain a good understanding of many of these areas during this time

17 Glossary

ADSM - the ADSTAR Distributed Storage Manager - a storage management product from IBM

AFS - the Andrew (distributed) filesystem

CORBA- the Common Object Request Broker Architecture from the OMG

CORE - Centrally Operated Risc Environment

Data mining the efficient discovery of previously unknown relationships and global patterns in large databases

DFS - the OSF/DCE distributed filesystem based upon AFS

DMIG - the Data Management Interface Group

GB - 10^9 bytes

HPSS High Performance Storage System - a high-end mass storage system developed by a consortium consisting of end-user sites and commercial companies

KB 2^{10} (1024) bytes - normally referred to as 10^3 bytes

IEEE the Institute of Electrical and Electronics Engineers

MB - 10^6 bytes

MSS a Mass Storage System

NFS the Network Filesystem developed by Sun

ODBMS - an Object Database Management System

ODMG the Object Database Management Group who develop standards of ODBMSes

OMG the Object Management Group

OQL the Object Query Language defined by the ODMG

ORB an Object Request Broker

OSM Open Storage Manager: a commercial MSS (Computer Associates)

PB 10^{15} bytes

SQL Standard Query Language: the language used for issuing queries against (relational) databases

STL The Standard Template Library - a component of the (draft) standard C++ library supporting collections etc

SSSWG the Storage System Standards Working Group A sub group of the IEEE Computer Society Mass Storage Committee

TB 10^{12} bytes

VLDB Very Large Database

VMLDB Very Many Large Databases

XBSA - the draft X/Open Backup Services Application Program Interface

18 References

[1] Object Data Management R G G Cattell, Addison Wesley, ISBN 0-201-54748-1

[2] DBMS Needs Assessment for Objects, Barry and Associates (release 3)

[3] The Object-Oriented Database System Manifesto M Atkinson, F Bancilhon, D DeWitt, K Dittrich, D Maier, and S Zdonik In Proceedings of the First International Conference on Deductive and Object-Oriented Databases, pages 223-40, Kyoto, Japan, December 1989 Also appears in [4]

[4] Building an Object-Oriented Database System The Story of O_2 F Bancilhon, C Delobel, and P Kanellakis (eds) Morgan Kaufmann, 1992

[5] Object Oriented Databases Technology, Applications and Products Bindu R Rao, McGraw Hill, ISBN 0-07-051279-5

[6] Object Databases - The Essentials, Mary E S Loomis, Addison Wesley, ISBN 0-201-56341-X

[7] Modern Database Systems - The Object Model, Interoperability and Beyond, Won Kim, Addison Wesley, ISBN 0-201-59098-0

[8] Objets et Bases de Donnees - le SGBD O_2, Michel Adiba, Christine Collet, Hermes, ISBN 2-86601-368-9

[9] Object Management Group The Common Object Request Broker Architecture and Specification, Revision 1 1, OMG TC Document 91 12 1, 1991

[10] Object Management Group Persistent Object Service Specification, Revision 1 0 OMG Document numbers 94-1-1 and 94-10-7

[11] The Object Database Standard, ODMG-93, Edited by R G G Cattell, ISBN 1-55860-302-6, Morgan Kaufmann (publishers)

[12] "Object Databases and Mass Storage Systems - the Prognosis ' - the RD45 collaboration, CERN/LHCC 96/17, CERN, 1996

[13] "Storage - At the Forefront of Information Infrastructures" - Proceedings of the 14th IEEE Symposium on Mass Storage Systems, September 1995

[14] "Database Computing in High Energy Physics", Drew Baden (University of Maryland) and Bob Grossman (University of Illinois at Chicago), in Proceedings of CHEP '91, Tsukuba, Japan, March 1991, Universal Academy Press

[15] "Databases for High Energy Physics ', D Baden, B Linder, R Mount, J Shiers, in Proceedings of CHEP '92, Annecy France, September 1992, CERN 92-07

[16] "Database Computing in HEP - Progress Report", in Proceedings of CHEP '92 Annecy France, September 1992, CERN 92-07

[17] "The PASS Project A Progress Report ', in Proceedings of CHEP '94, San Francisco US, April 1994, LBL-35822

[18] "Object Operations Benchmark", Cattell, Skeen, ACM Transactions on Database Systems, April 1992

[19] "The OO7 Object-Oriented Database Benchmark , Carey, DeWitt, Naughton, in Proceedings of the ACM SIGMOD Conference, Washington D C , May 1993

[20] "Software Engineering Standards", C Mazza, J Fairclough, B Melton, D De Pablo, A Scheffer and R Stevens, Prentice Hall, ISBN 0-13-106568-8

[21] "Software Engineering Guides", C Mazza, J Fairclough, B Melton, D De Pablo, A Scheffer, R Stevens, M Jones and G Alvisi, Prentice Hall, ISBN 0-13-449281-1

[22] See http //www xopen org/public/pubs/catalog/p424 htm Also available as X/Open Preliminary Specification P424 ISBN 1-85912-056-3

[23] From VLDB to VMLDB (Very MANY Large Databases), Stuart E Madnick - Proceedings of the 21st VLDB Conference, Zurich, Switzerland, 1995 (invited talk)

[24] Predictions and Challenges for Database Systems in the Year 2000, Patricia G Sellinger, IBM Almaden Research Centre Proceedings of the 19th VLDB Conference, Dublin, Ireland, 1993