DAD – Distributed Adamo Database system at Hermes

W. Wander, M. Düren, M. Ferstl

Universät Erlangen-Nürnberg, Physikalisches Insititut,
Erwin-Rommel-Str. 1, D-91058 Erlangen Germany

K. Ackerstaff, M. A. Funk

DESY, Notkestrasse 85
D-22603 Hamburg, Germany

P. Green

University of Alberta Centre for Subatomic Research, Department of Physics
Edmonton, Alberta T6G 2N5, Canada

Ph. Oelwein

MPI für Kernphysik,
Postfach 103980, D-69029 Heidelberg, Germany

D. Potterveld

Argonne National Laboratory, Physics Division, Building 203,
9700 South Cass Avenue, Argonne, Illinois 60439, USA

P. Welch

Department of Physics, Oregon State University
Weniger Hall 301 , Corvallis, Oregon 97331-6507, USA

Software development for the HERMES experiment faces the challenges of many other experiments in modern High Energy Physics: Complex data structures and relationships have to be processed at high I/O rate. Experimental control and data analysis are done on a distributed environment of CPUs with various operating systems and requires access to different time dependent databases like calibration and geometry. Slow- and experimental control have a need for flexible inter-process-communication. Program development is done in different programming languages where interfaces to the libraries should not restrict the capabilities of the language. The needs of handling complex data structures are fulfilled by the ADAMO[2] entity relationship model. Mixed language programming can be provided using the CFORTRAN[3] package. DAD, the Distributed ADAMO Database library, was developed to provide the I/O and database functionality requirements.

1 Introduction

The initial goal for the DAD project was to provide detector calibration and geometry data for any given time period of the HERMES-experiment in a central database server. HERMES is a high statistics high energy experiment at the HERA-ring at DESY, with the aim to measure the spin structure functions of protons and neutrons, by scattering a polarised electron beam from a polarised internal gas target. As HERMES-software is relies on the ADAMO[2] entity relationship model for data handling, an approach based on ADAMO seemed to be suitable. During the development phase, DAD was extended to a message passing system to fulfill needs for an ADAMO based slow control system and by a fast machine independent event stream in an SMP environment. Together with PINK[1] – a DAD-tk/tcl extension – DAD is now extensively used in the HERMES slow-control and analysis software.

2 The Client Server Model

As our software development profits from the benefits of the ADAMO model – e.g. separating data description from source code and thus improving the documentation and maintainability of data streams and software; e.g. including the data definition into the data-stream and thus keeping data accessible across version changes of both software and data models; e.g. providing a standard data access for both *C* and FORTRAN code, …– it also suffers from the ADAMO underlying memory management and I/O capabilities. Whereas a flexible and more modern memory management is now available in the ADAMO smalltap extension, e.g. random access to database files is neither provided to multiple processes at the same time nor is it possible to access database information from different systems of a computing cluster.

Software analysis on SMP architectures and/or on analysis farms has to overcome these restrictions. Therefore information providers, called servers, have to be included between direct file I/O and the data processing tasks, called clients (see fig. 1).

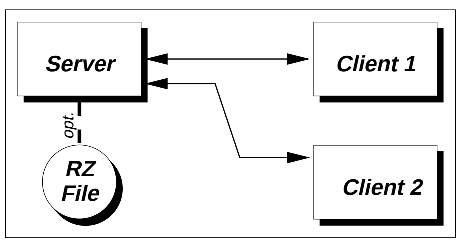

Figure 1: A simple client server model for DAD.

A client connecting to a DAD-server accesses a so called *dataflow*. A *dataflow* is an ADAMO object which combines other objects like *tables* and *relationships* among them (see fig. 2).

The first information the client then receives is a description of the server's data definition representation of this dataflow. The client then either uses this information to generate an equivalent data model, like PINK[1] or HEP[4], or matches it against its own existing data description. This guarantees an information exchange even if the client's definition does not exactly match the server's definition.

DAD distinguishes between three different operation modes:

- **Information exchange on record basis.**
  A record is a complete set of information filled into the tables and relationships of a

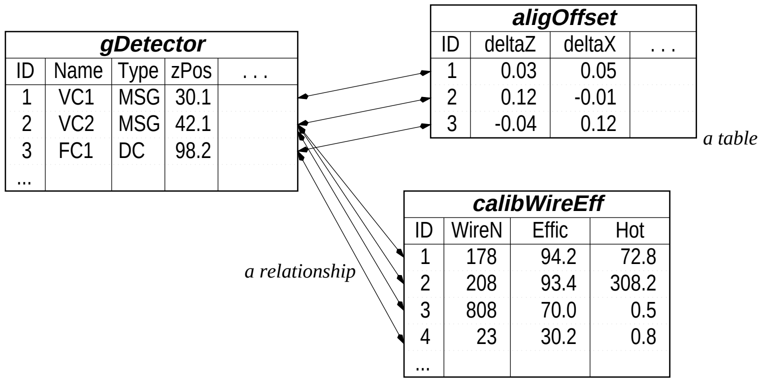

Figure 2: An example of an ADAMO dataflow.

dataflow. This for instance can be an event of the data-stream, a geometry description or a lookup table. Clients can read these records, select them according to specified rules, change, write and generates them.

- **Access of time dependent information.**
  Time dependent databases are of special interest for the data analysis. Calibration, alignment and efficiencies as well as the mapping of read-out channels are varying in time . DAD provides easy calls to access and generate data sets with a limited validity period. For the client these data sets do not differ from the above mentioned records – however the storage and data transmission between client and server does only concern changed entries.

- **Booking of information.**
  A requirement for slow- and experimental control applications is inter-process-communication, in a distributed computing environment this *IPC* should of course not be restricted to a single system. DAD can provide these requirements by introducing a booking scheme where clients can book new information on the server. This information may be a command table, where clients only book information which is addressed to them and thus can talk to each other via the server, or any other kind of data like hardware status and beam information for monitoring purposes.

With the above mentioned restriction of only one process accessing a random access ADAMO file at a time, DAD-servers cannot simply fork like most other tcp/ip servers do, when a new connection is requested. However this restriction turned out to become an advantage for the DAD concept as it avoids the heavy system load of large processes forking at high rates like e.g. most WWW servers do and thus DAD-servers can easily handle several thousand requests per minute. This single process technique enforces a multi-threaded event handling in the server to avoid slow or blocking clients from blocking the connections to other clients of a server. Figure 3 illustrates this concept.

Another important feature of DAD-servers is the authentication scheme. For obvious reasons write access to the experimental control should not be granted to the whole Internet

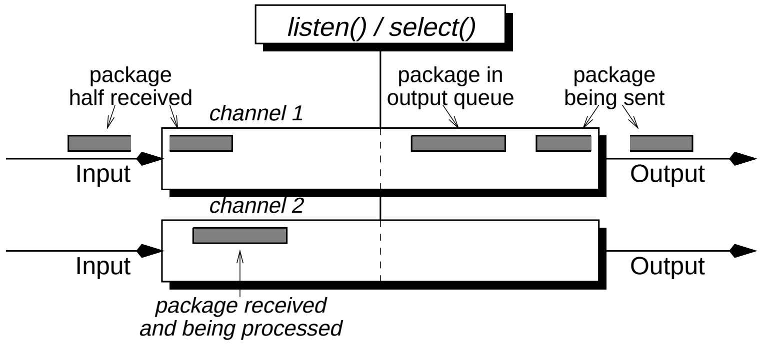

Figure 3: The multi-threaded I/O concept. While packages being received on some channels, others might be used to send pending output packages or to process completely received packages.

community. Therefore authentication can be both host based or user/password based. Passwords are exchanged encrypted on the net.

3 DAD sequential I/O

In addition to the I/O to and from the servers, DAD streams can also be used to access files and pipes in DAD format. The advantage in comparison to the standard Zebra FZ drivers of ADAMO is the improved speed, the reduced data volume and the increased flexibility for data definition changes. Additionally, DAD pipes can be used to distribute data records to different processes on the same system, enlarging the throughput in SMP environments (see fig. 4).

DAD also provides tools for filtering and manipulating data-streams in DAD- or ADAMO-format. The program HEXE can run as a filter like the following example demonstrates:

```
$ hexe hrc.output --output hexe.output \
       ++filter     'rcCluster:E > 1.5' \
       --expression 'rcTrack:2:rProbPion>0.8'
```

Here the input pipe is represented by the file `hrc.output`, the output pipe is named `hexe.output`. Calorimeter cluster information is only kept in the stream if the cluster energy is higher than 1.5 GeV. Records (here: events) are only kept in the stream, if they contain at least two pion tracks.
HEXE itself does not know about the HERMES data representation. It generates it from the input stream, therefore it is a useful generic tool for the whole ADAMO community.

4 DAD and ADAMO in the HERMES-Software

Currently DAD is extensively involved in the HERMES analysis software. It has been ported to IRIX 5.3, OSF1, Ultrix 4.3, SunOS, Linux and even VMS, other systems are in preparation by non-HERMES groups. Figure 4 gives a simplified overview over the current analysis chain design where DAD is used for both the event stream and the communication with the database on the various servers. The consequent use of both DAD and ADAMO also led to the successful implementation of the different software parts: Real physics data were available three hours after the HERMES detector was turned on for the first time. And though data modelling tools, like ADAMO, always have an impact on CPU consumption the reconstruction time is well below 30ms per event, thus HERMES can even afford to run two productions in parallel keeping up with the data taking.

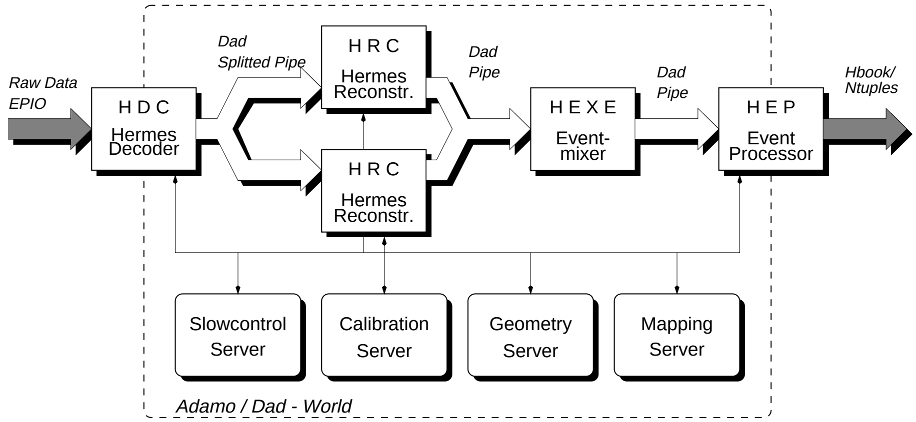

Figure 4: The HERMES event stream. Data is fed into the decoder in EPIO format where it is decoded and calibrated and sent to the Reconstruction programs. The events are then filtered by the HEXE-Program (part of DAD) and sent to the event processor to form histograms and ntuples for further analysis.

Slow- and experimental control are even more involved in the DAD scheme. Up to one thousand client connections are often established to four different servers. They control hardware interaction, monitoring, checking and archiving functions. Here specially PINK[1] and its derivatives are in operation and provide an easy programmer- and user- interface to DAD and ADAMO data.

References

1. K. Ackerstaff and M.A. Funk, *A TCL/TK based Database Interface to ADAMO and DAD*, WWW: http://dxhra1.desy.de/˜maf/pink/pink.html, Proc. of CHEP 95, Rio de Janeiro, 18.-22. Sept. 95, to be published.
2. *ADAMO – Entity-Relationship Programming System, Version 3.3*, WWW: http://www1.cern.ch/Adamo/ADAMO_ENTRY.html, Programming Techniques Group, ECP Division, CERN, Geneva, 1994.
3. Burkhard D. Burow, *Mixed Language Programming*, Proc. of CHEP 95, Rio de Janeiro, 18.-22. Sept. 95, to be published.
4. M. Ferstl, *HERMES Event Processor Manual, Version 2.1*, WWW: http://dxhra1.desy.de/documentation/hepmanual/hepmanual.html, HERMES Collaboration, 1995.
5. W. Wander, DAD – an Adamo based distributed Database concept, WWW: http://dxhra1.desy.de/dad, HERMES Collaboration, 1995.