

# **DAD – Distributed Adamo Database system at Hermes**

W. Wander  
HERMES-Collaboration  
Universität Erlangen Nürnberg  
Physikalisches Institut  
Erwin-Rommel-Str. 1  
D-91058 Erlangen  
Germany

18-22. Sep. 1995

# Outline

- Introduction
- Adamo is the Base for HERMES Software
- Client-Server Concept
- DAD Sequential I/O
- The DAD Library
- DAD in the HERMES Environment

# Introduction

High Energy Physics data analysis requires (among other topics).

- Access to complex data structures and data relationships.
- Access to common databases from a variety of computer systems.
- Fast & flexible I/O drivers.
- Interprocess communication across platforms for slow and experimental control.
- User definable filters for event and data streams.

⇒ DAD: Distributed Adamo Database

## ADAMO is the Base

for the whole HERMES software  $\implies$  also for DAD.

- ADAMO<sup>1</sup> provides easy access to data structures and relationships of the HERMES data sets.
- ADAMO separates code from data definition  $\implies$  improved documentation and maintainability.
- ADAMO (+ CFORTRAN<sup>2</sup>) allows FORTRAN & C access to the data.

### A DDL example

```
SpacePoint =  
(  
  X = REAL   : 'X coordinate',  
  Y = REAL   : 'Y coordinate',  
  Z = REAL   : 'Z coordinate',  
) : 'Space Point';
```

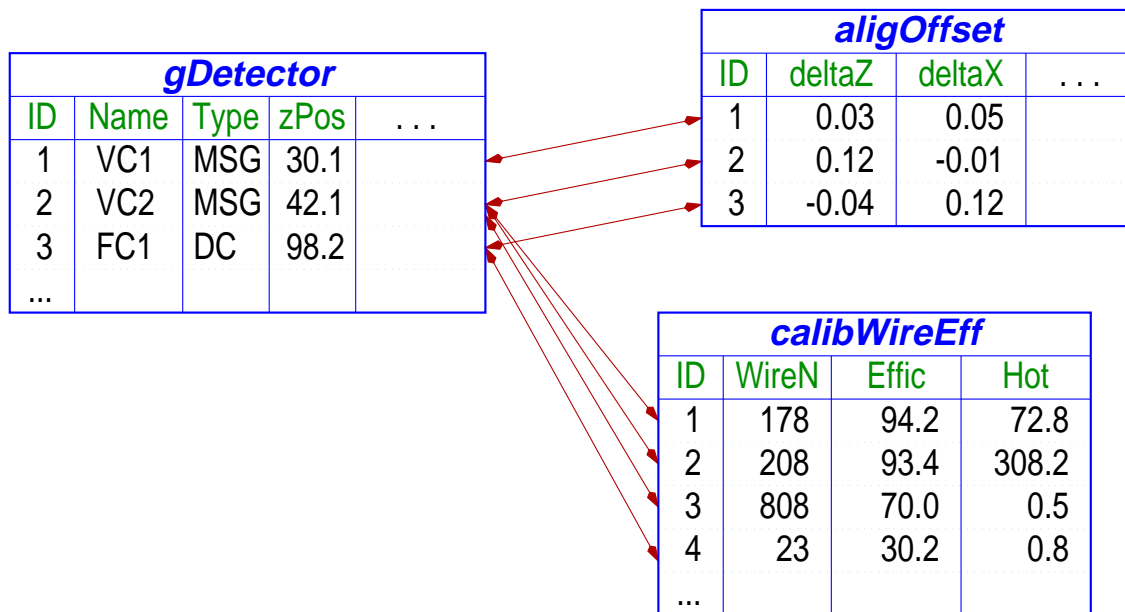
<sup>1</sup>developed by PTG CERN

<sup>2</sup>see talk by B.D. Burow

ADAMO however also has some drawbacks:

- It's memory management relies on ZEBRA (not true for smalltap).
- I/O is also based on ZEBRA  $\implies$  no shared write access to common database files.
- No security or recovery of data base files after power failures available.

# ADAMO Basics

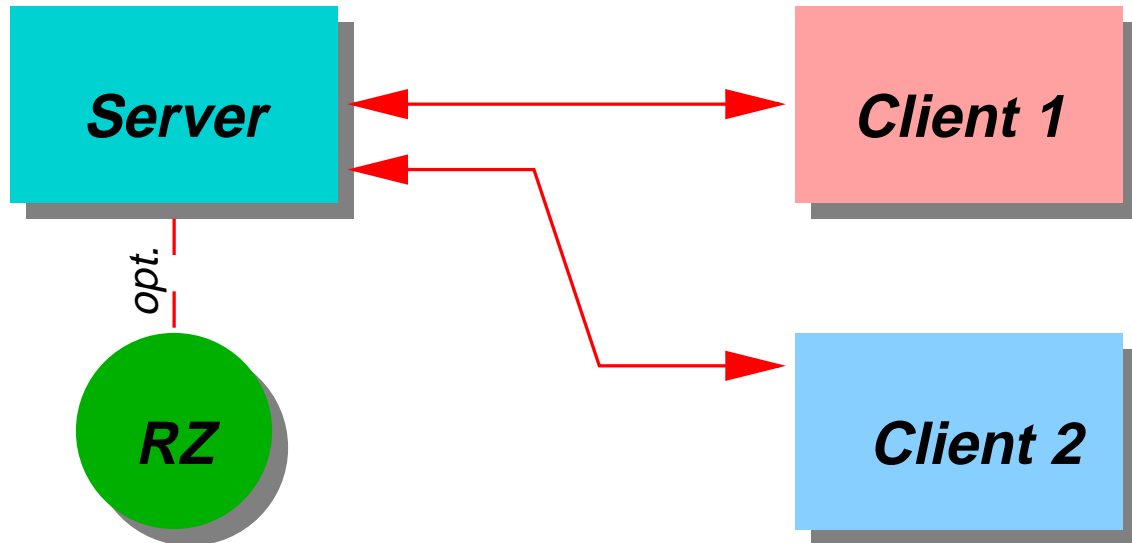


Data access is strictly object based. Objects for ADAMO are

- **Tables:** Rows of data sets.
- **Relationships:** Relationships of different type between Tables.
- **Dataflows:** Sets of Tables and Dataflows.
- ...

ADAMO provides controlled access to it's objects and a large set of data processing calls  
⇒ secure programming environment.

## DAD's Client-Server-Concept



Clients connect to the server via TCP/IP on INET or UNIX sockets and  
– we have learned from ADAMO–  
exchange their data definition.

⇒ A change in the data definition does not affect the system.

⇒ Generic tools can act on the data (HEXE, HEP, PINK<sup>3</sup>).

---

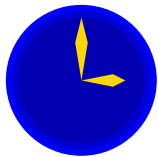
<sup>3</sup>see talk by K. Ackerstaff

# Operating Modes

DAD client server connections know three basic operating modes:



Operation on Records. i.e. the data of a complete dataflow.

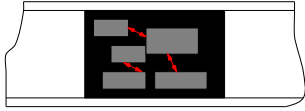


Access to time dependent information.



Booking information as it changes on the server.

## Operation on Records



A record is the data filled into a dataflow, (into the tables of a dataflow). Examples for records are

events , look-up tables , ...

Records can be transmitted as a whole to and from the server.

Server records can be selected for certain conditions.

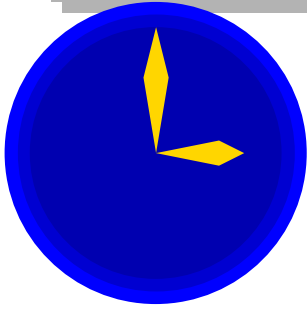
```
Event.TriggerMask & 12 = 8
```

Single tables or selected rows of tables can be transmitted.

```
dadGetTabIf(Id, Drift,  
            "Drift.rPuls < 2.5", err);
```

Semaphores control the access to prevent inconsistencies from multiple write accesses.

## Time Dependent Information



...is needed as

⇒ Calibration constants, alignment, efficiencies, even mapping and slow-control information vary with time.

...and analysis requires this information for each data taking period.

DAD provides:

- calls to receive data for a given time period.
- calls to store data for a given time period.
- compressed storage and I/O for time dependent information as only changes concern.

⇒ Start the analysis chain on any run and you will get correct calibration data.

## Time Dependent Information

An example for updating the calibration data:

```
...
dadInit();
dadConnect( "Calibration", &CalibID, &err);
/* forget the error checking
   (only for CHEP!) */
dadFtoTime( 950528, 123000, &UnixTime);
dadUpdate( CalibId, UnixTime, true, &err);
dadDisconnect( CalibId, &err );
...

/* we could do it in f77 ... but
   we NEED NOT ;-) */
```

## Booking



Booking provides clients with latest information.

This is needed for

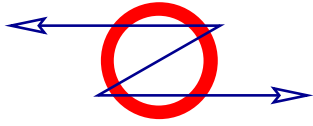
- Online Analysis – where latest calibration data must be fed into the production.
- Slow and experimental control – where inter process communication is required and new data is needed for monitoring and check tasks.

Clients book information they are interested in:

```
dadBookIf( CmdID, Commands,  
           "Receiver = 'client1'", &err);
```

As soon as the server receives a new table with `Commands.Receiver` equals to `'client1'` this will then be sent.

## Authentication



Experimental and slow control are not designed to be operated by any member of the Internet community.

⇒ Authentication required !

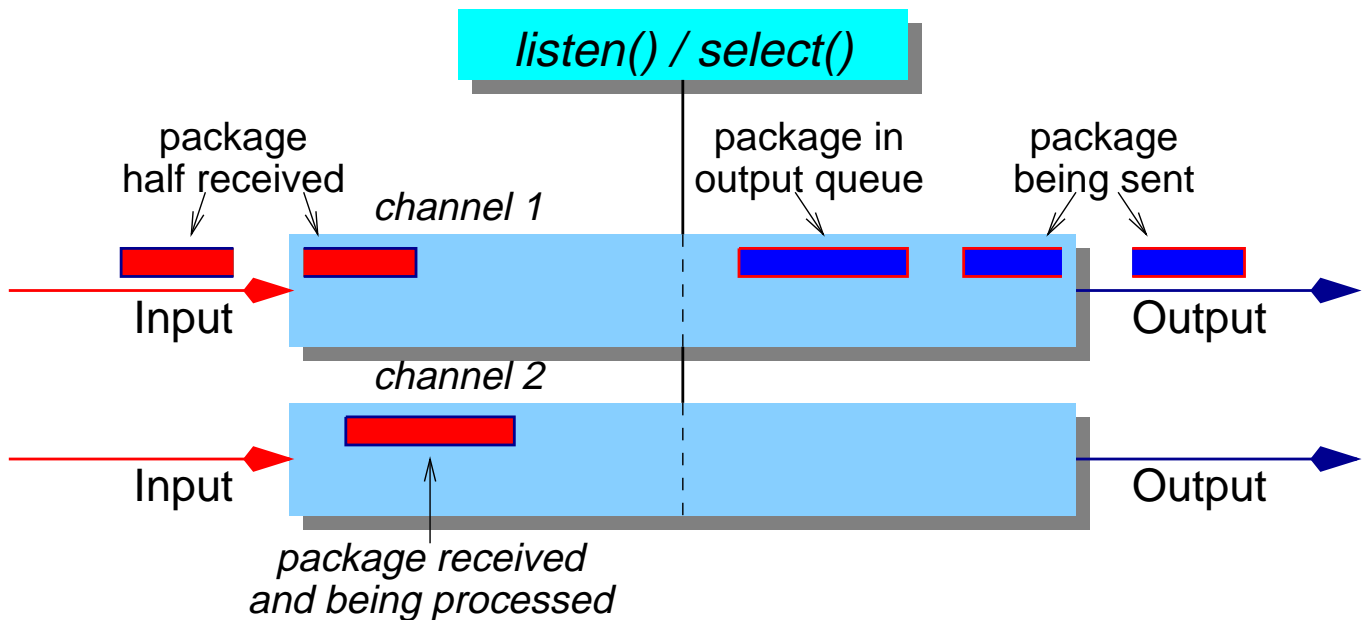
Servers can restrict read/write access to dataflows on a host or user level.

Commands to maintain DAD-passwords are provided.

## Server I/O

DAD servers operate on Zebra RZ files  
⇒ forking the server for each new connection  
– like e.g. most WWW servers do – is impossible.

This results also in higher throughput and less system load if server kernel is multi-threaded:

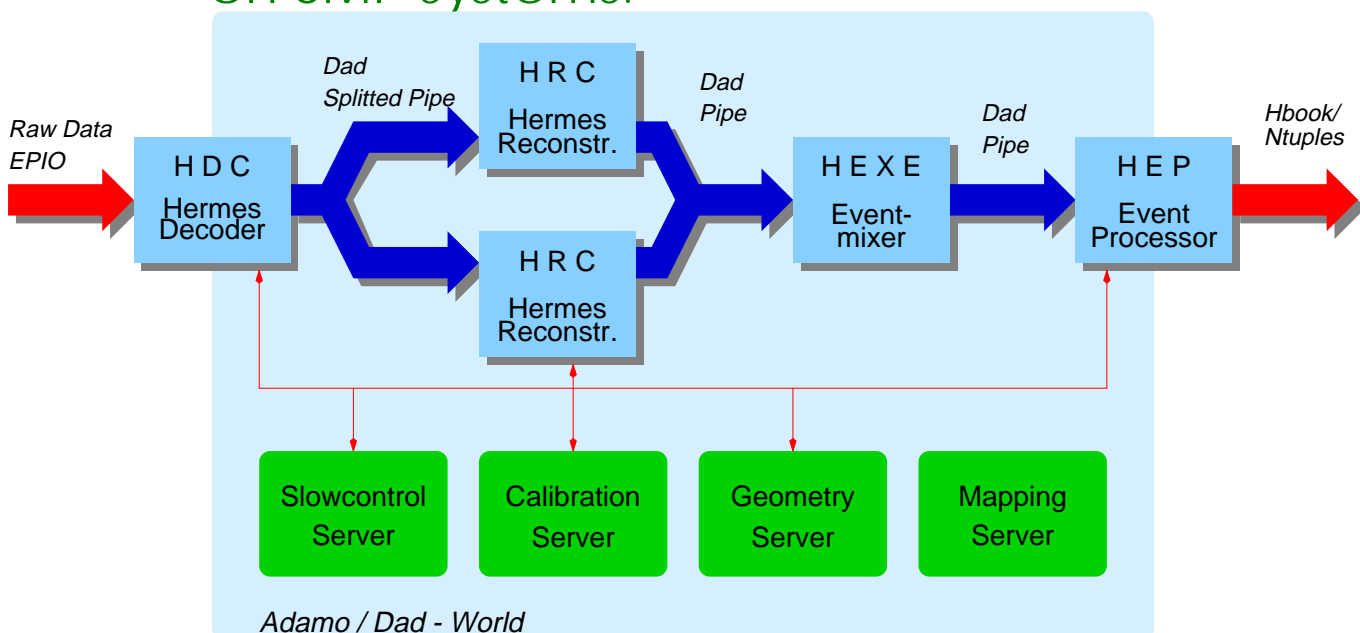


⇒ More than several 1000 accesses per minute possible.

## Sequential I/O

ADAMO's sequential file I/O-drivers exist and are used in many applications. Why does DAD need it's own drivers?

- DAD client-server I/O exists.
- No ZEBRA drivers  $\Rightarrow$  improved speed.
- Even increased flexibility over the ADAMO data definition matching.
- Support for pipes and splitted pipes available.  
 $\Rightarrow$  Splitting the data stream for parallel processing on SMP systems.



# HEXE

the DAD record / event filter.

HEXE is a generic tool to filter data streams in both ADAMO and DAD format.

HEXE reads its data definition from the event stream and parses user filter expressions:

```
# hexe hexe.input --output hexe.output \  
> +f 'Cluster:Energy > 1.5' \  
> -e 'Track:2:ProbPion > 0.8'
```

will read from `hexe.input` and write to `hexe.output` writing only those events which contain at least 2 (charged)  $\pi$  Tracks and removing all calorimeter clusters of less than 1.5 GeV from the event.

## The DAD-library

DAD has been ported to IRIX, OSF, Ultrix, SunOS, Linux and even VMS (Other ports in progress). `configure` scripts available.

Contains about 21000 lines of C source code and 342 lines of FORTRAN code.

DAD provides 60 user callable and documented functions.

DAD documentation is available on <http://dxhra1.desy.de/dad/> or in a 130pages postscript document also available there.

Many generic tools for DAD available: FLOYD, HEXE, HEP, PINK<sup>4</sup>, ...

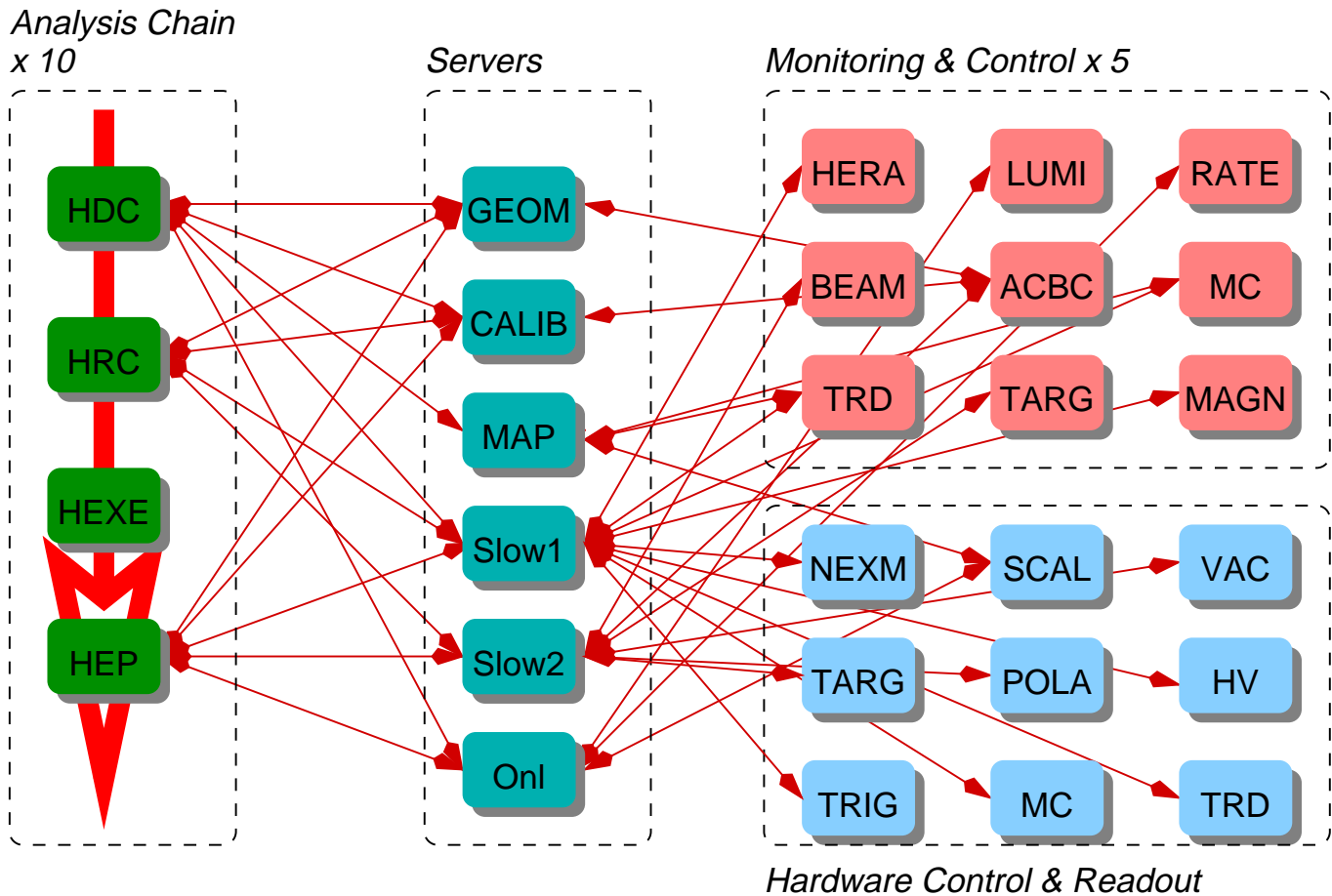
DAD is under a severe test by the HERMES software and provides stable operation conditions:

---

<sup>4</sup>see talk by K. Ackerstaff

# HERMES Reality

Well – part of it. . .



. . . and it is still growing.

## Conclusions

Consequent use of ADAMO and DAD in the HERMES Analysis Software also contributed to its success:

First physics data was available three hours after the complete detector was switched on for the first time.

And we are **not** slowed down by the tools: Full reconstruction runs at a rate of 20–30Hz per single SGI CPU.

⇒ Enjoy the next colorful talk about PINK<sup>5</sup>

---

<sup>5</sup>see talk by K. Ackerstaff