Distr.: SOFTWR

ALEPH 86-20
24 January, 1986
A.J. Flavell

MINUTES OF THE ALEPH OFFLINE SOFTWARE MEETING
CERN, 20 NOV 1985

Present: BARDADIN M. BELLIARD M. BERTHON M. BEUSELINCK R. BLOCH-DEVAUX B. BLONDEL A. BOCK R. BOSMON M. BOUCROT J. BOUISSEUT A. BOWDERY C.K. BRAUN O. BUNN J. BURKIMSHER P.C. CAPON G. CLIFFT R.W. CRESPO J.M. DEBOUARD G. DEMILLE D. DEPALMA M. DORNAN P.J. DREVERMANN H. FERNANDEZ E. FINCH A.J. FISHER S.M. FLAVELL A. GREEN M.G. GRIVAZ J.F. HAGELBERG R. HANSEN J.D. HEIDEN M. IASELLI G. KELLNER G. LAI C. LANDON M. MALLET V. MANNER W. MCCLATCHEY R. MERMIKIDES M. NEUGEBAUER E. PALAZZI P. PALLIN D. PIJLGROMS B. PUN T. PUTZER A. QIAN Z. RASO G. RENARDY J.F. REYROLLE M. SEDGBEER J.K. STIMPFL G. STUPPERICH K. TALEBZADEH M. THOMPSON J.C. TROST H-J. VIDEAU H. WANG S. WANG T. ZHAO W. ZITO G.

---

1. TOWARDS A PHYSICS ANALYSIS WORKSTATION - R.BOCK

Traditional physics analysis uses HBOOK to manipulate data from a DST and produce paper output and/or sequential files. Publications are produced in typing pool and drawing office. The concept of a Physics Analysis Workstation is proposed, to exploit present and future workstation hardware. Physics analysis would be done interactively, building up a database of results, and finally producing publication-quality plots. Facilities are required in the areas of Data Object handling, Graphics, and Command handling; the extent to which existing packages provide the necessary facilities was considered.

Discussion. Writing papers at the terminal with graphics in - exists already (2D graphics) at DESY - would be investigated. Interleave on the Apollo also does it - very expensively! Discussion of HBOOK4 facilities - will be compatible with present HBOOK, some additions. Any idea what the 'Micro DST' will be - yes, it is an n-dimensional histogram stored by entries, kept as a disk file, going to MSS/tape when too much.

2. PHYSICIST REQUIREMENTS FOR INTERACTIVE PROCESSING/DISPLAY - H.VIDEAU

What do you want to do sitting in front of your screen: Run processes on data, choose, control, modify, interrupt; look at results, event display, (statistics); changes - to the data (omit a point and reprocess), - to the process (modify some parameters and refit); keep a journal - enabling replay of all or part of a session. The ability to interrupt a process (stop it in a controlled fashion) is not common in user programs but is a very nice idea, although not part of the standard language so not fully portable. There are several implications for the structure of software (for this purpose! - batch might be very different). The Data Flow Diagram '99' implies a requirement for graphics items to be picked and then identified with items in the Event Store.

If the structure is right, it remains open to new packages and to new types of machine. People developing programs are asked to bear this in mind.

3. ADAMO TOOLS FOR SA/SD - PRESENTED BY P.PALAZZI

This was a collaborative effort including R.Brazioli, S.M.Fisher and W.Zhao. The Entity/Relationship model has been coupled with Structured Analysis. A four-way picture (slide 2) shows the relationships between Data and Process. By coupling the two approaches, the various data flows are not just descriptive names but specific things in the ER model. Blob 1 (slide 4, expanded on slide 5) was done at the ADAMO presentation: Adamo Data Dictionary, Conceptual Schema Processor, Interactive Query Processor. Blob 3 is presented now (expanded on slide 6). TekSA produces diagrams such as the example shown. TekDiagramDesc produces a machine-readable ASCII file equivalent to the diagram: converted to Data Flow language for internal reasons. DFP produces standard error reports, and stores meta tables in ADD (each blob is a VAX COM file). From 1.2 come also metafiles from ER analysis. SCP (3.5) is an isolated processor which draws a structure chart (not used internally) for documentation purposes.

The whole thing was set up by a bootstrapping method. Improvements could be made based on response and demands from the users. Some evaluation is being done of using a front-end processor (e.g. a Mac) for user dialogue, communicating with a back end processor (e.g. VAX) which manages the ADD.

4. STATUS OF EVALUATION OF COMMERCIAL TOOLS FOR SASD - G.KELLNER

The Tektronix SA tool has been installed at CERN and tried out for a lot of the ALEPH DFDs. Details were presented (see slides). The tool does not permit a composite dataflow from a high-level diagram to appear as several individual dataflows on the next lower level; an artificial 'datagate' has to be defined but it leads to messy looking diagrams as can be seen on the specimens. H.Videau has a workable solution but it brings problems of its own.

What else is available for SA - not much and rather restricted. The Tektronix SD tool is being developed but we cannot expect much in the forseeable future. What the ADAMO people did so far looks nice - see what it needs to produce a full SA-SD tool set.

Discussion. (HV:) If it is obvious who is the parent you don't need the data decomposition, but other times you need it for the checking. (PP:) The IQL will tell you that (but not by picking graphics, at least not yet!). (JK:) The Tek tools check many things but this can be seen as a stopgap solution. The ADAMO would be able to check everything which is in principle checkable. The PC tool looks convenient from the user point of view. (GK:) Confusing data flows of constants and associated data gates could be avoided by getting them from a data store instead.

5. AOB

People complained about the Reconstruction meetings being on Friday. Now that Monte Carlo requirements are reducing, Monday is suggested. B.Bloch agreed, provided that overlap can be avoided.

A volunteer was requested to take a turn as Minutes Secretary. After the meeting, Chris Bowdery of Lancaster offered to do this.

ORIGINS OF PAW

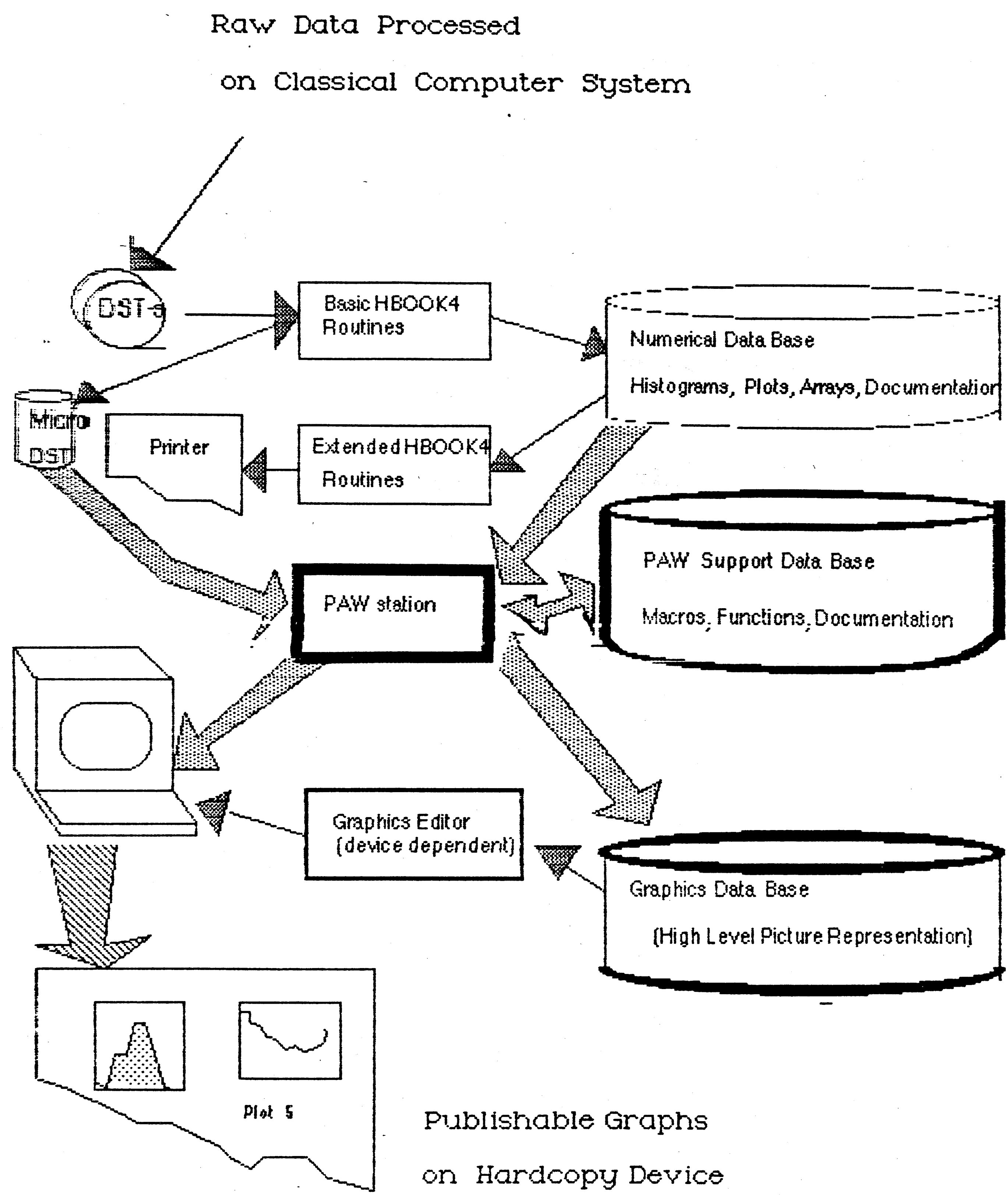
NOW

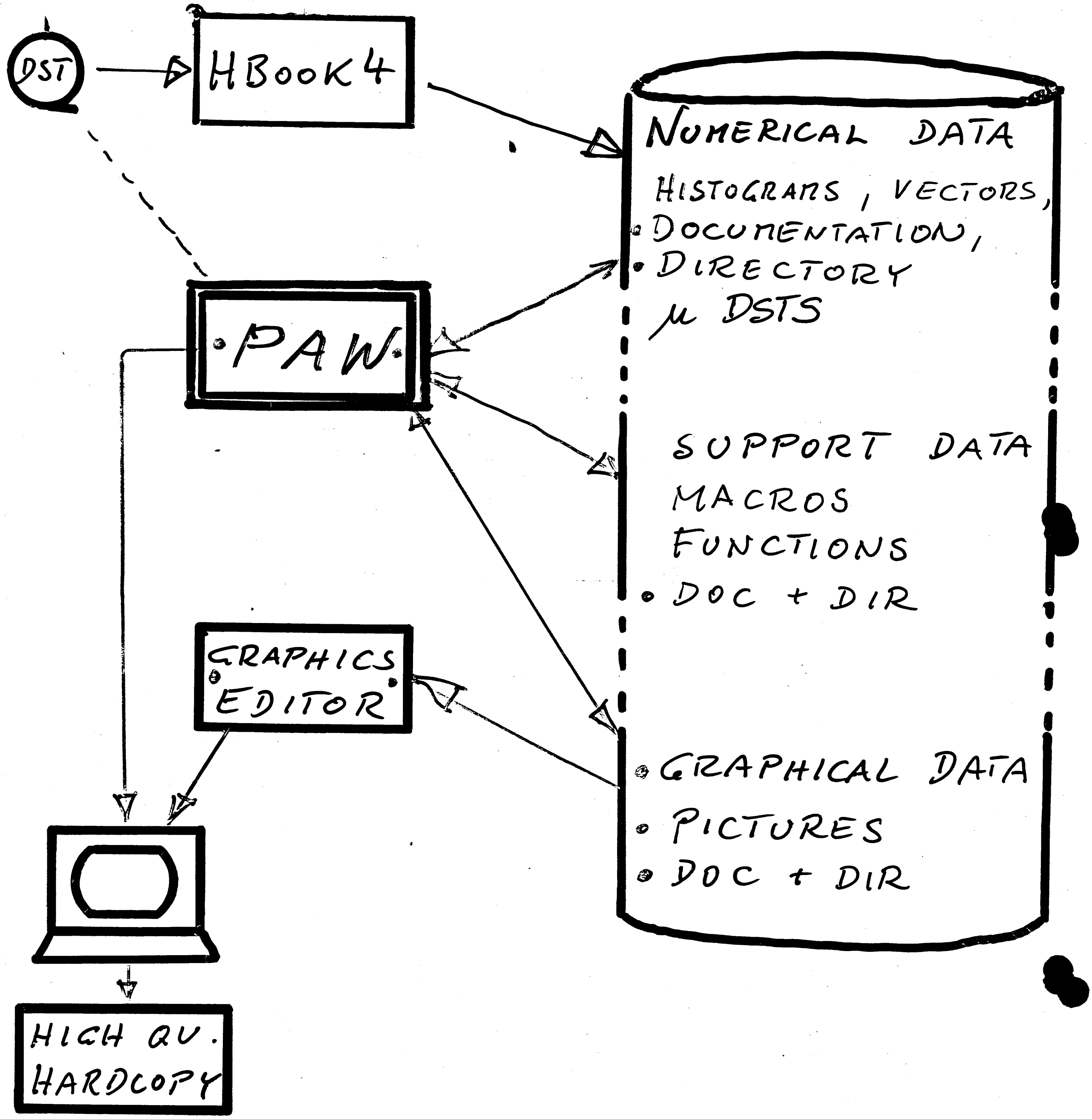
- ANALYSIS (STANDARD) ENDS WITH DST
- HBOOK PRODUCES PAPER AND/OR SEQU. FILES
- PUBLICATIONS ARE PRODUCED IN TYPING POOL AND DRAWING OFFICE
- WORKSTATIONS / PERSONAL COMPUTERS NOT GENERALLY USEABLE

PAW

- MAKES HBOOK / HPLOT INTERACTIVE
- DEFINES A 'DATA BASE' FOR RELEVANT OBJECTS (PLOT CONTENTS, COMMAND FILES, VECTORS, PICTURES)
- ALLOWS ADAPTATION TO FUTURE CONFIGURATIONS
- ENDS IDEALLY IN PUBLICATION - QUALITY PLOTS
- PROPOSES A STANDARD FOR HIGHLY SELECTED / COMPRESSED MICRO - DST-S-

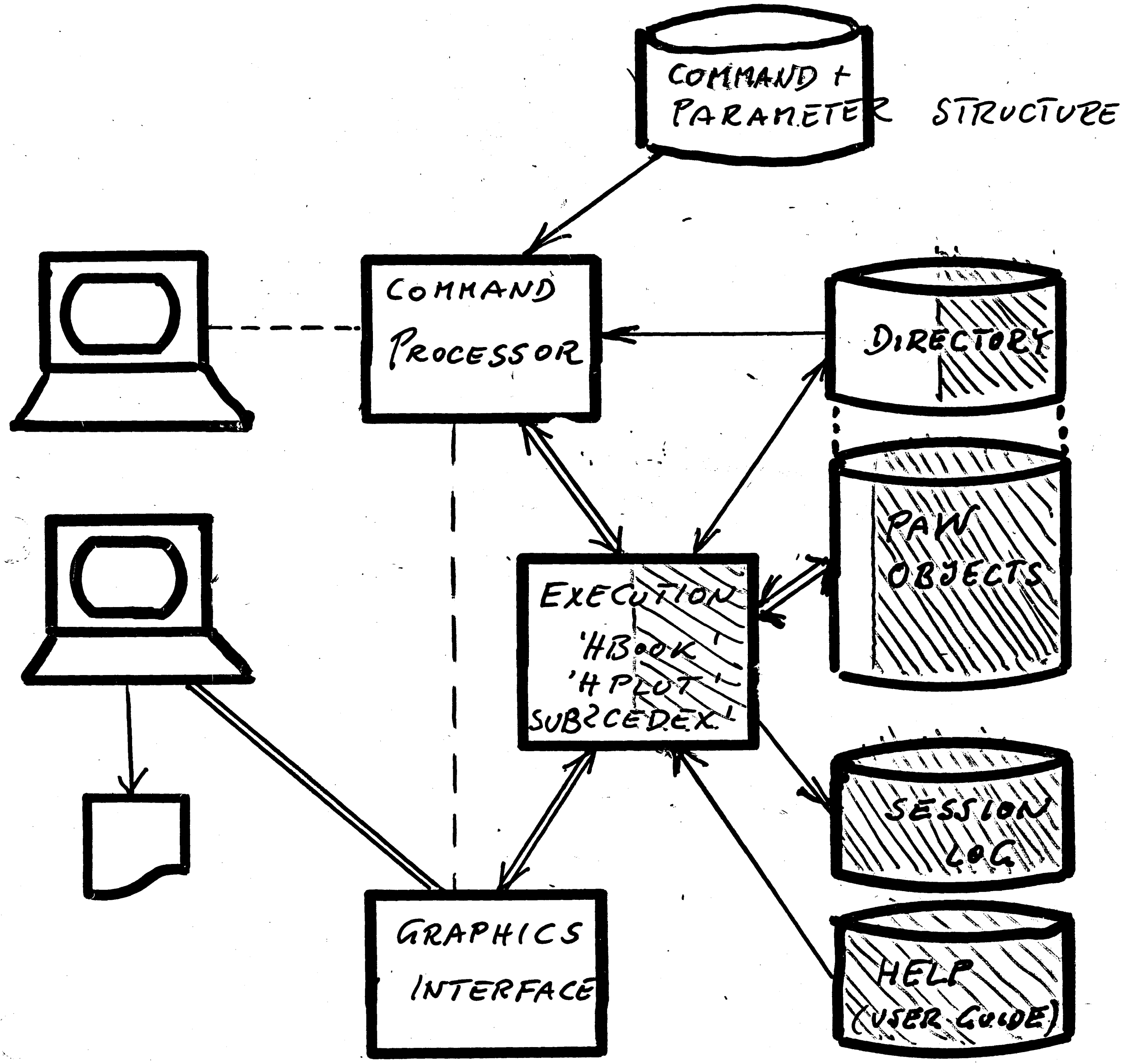
5 System Overview





PHYSICS ANALYSIS WORKSTATION

OVERVIEW



PAW

COMPONENTS

HOST - ORIENTED

SO FAR IDENTIFIED AREAS OF ACTIVITY

- GENERALIZED COMMAND INPUT
  TYPING / MENU / GUIDANCE

  DELPHI

- RE-THOUGHT GRAPHICS INTERFACE
  FROM TV TO INTERACTION AND
  3D

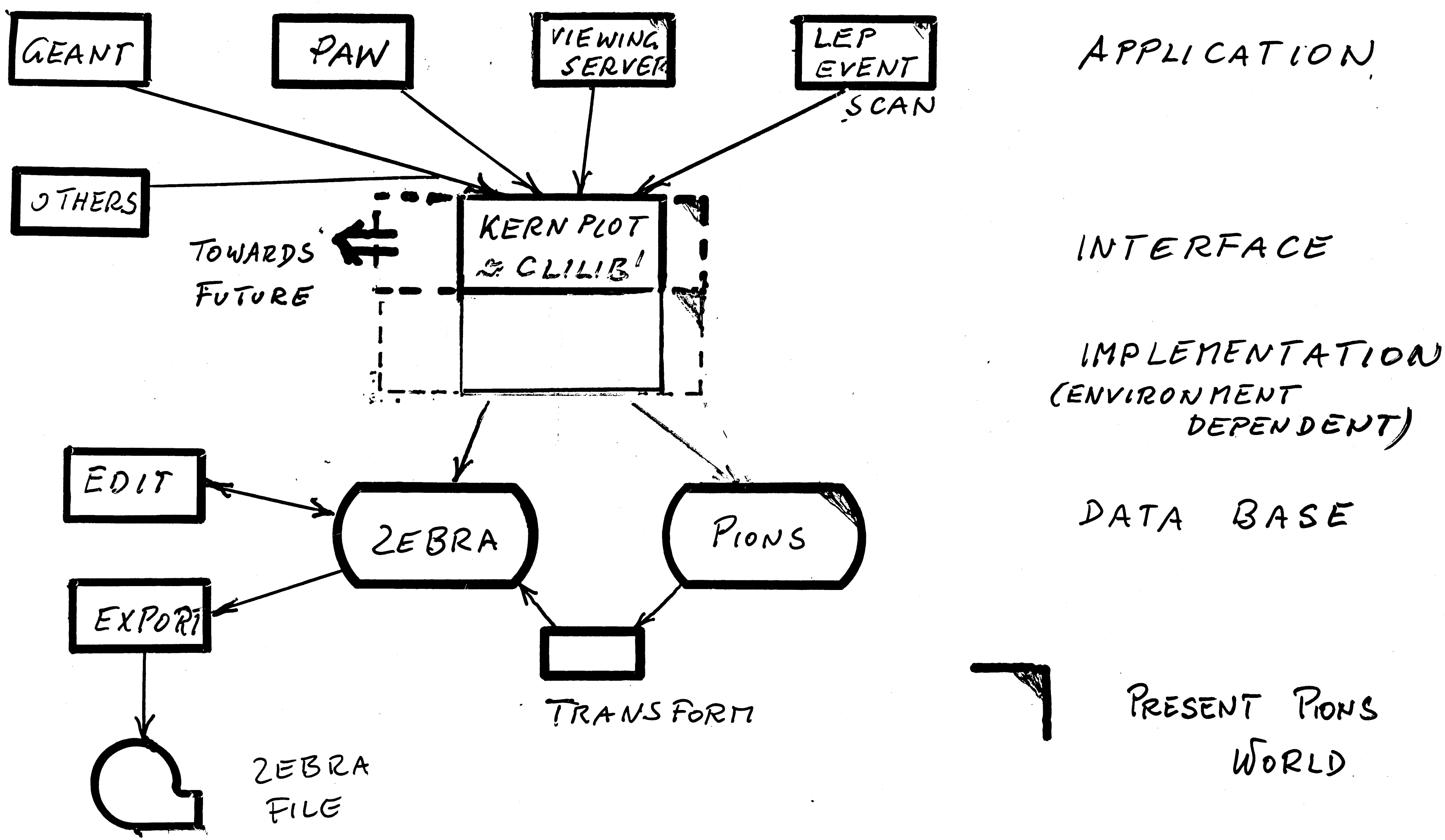
  MEDDLE

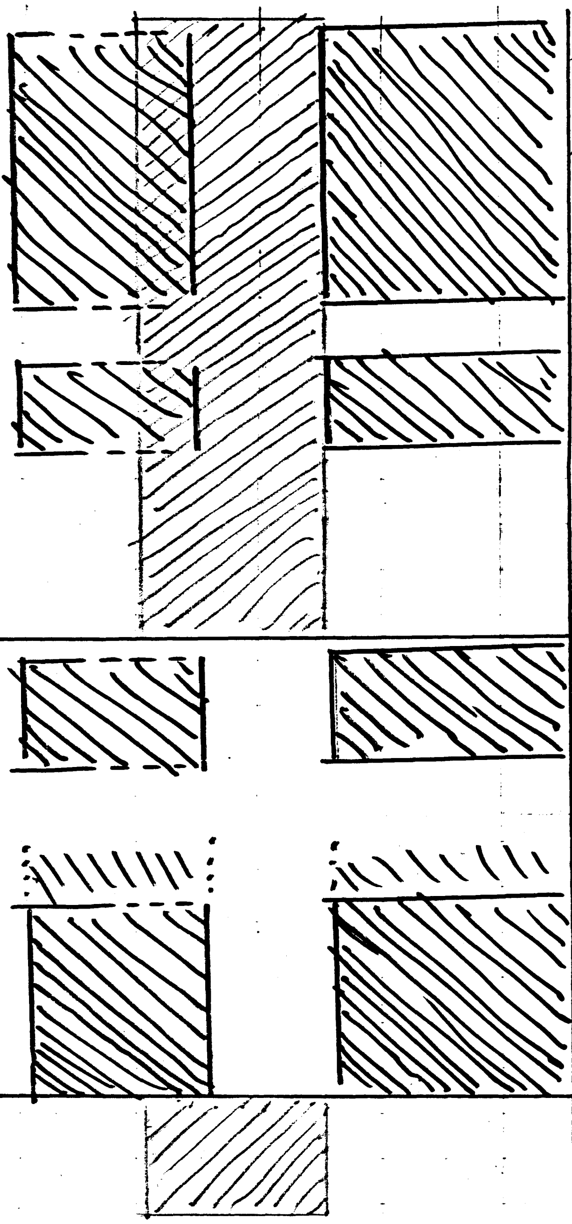
- DIRECT ACCESS DATA OBJECTS
  USING THE ZEBRA - RZ ACCESS
  METHOD, A PAW NAMING CONVENTION
  AND DOC POSSIBILITIES, → EXPORT
  HBOOK 4, ZEBRA

- OBJECT AND COMMAND EXCHANGE
  OVER LINES AND NETWORKS
  RS232 ? RPC ??

- GRAPHICS EDITING
- EMBEDDING GRAPHICS IN TEXT
- INTERACTIVE HISTOGRAMMING

} FUTURE


GEANT
PAW
VIEWING SERVER
LEP EVENT SCAN
OTHERS
TOWARDS FUTURE
KERN PLOT & CLILIB
EDIT
ZEBRA
PIONS
EXPORT
TRANSFORM
ZEBRA FILE
APPLICATION
INTERFACE
IMPLEMENTATION (ENVIRONMENT DEPENDENT)
DATA BASE
PRESENT PIONS WORLD



x PRIMITIVES 2D
PRIMITIVES 3D
x PRIMITIVE ATTRIBUTES
x INPUT (STRING, PICK, LOCATE ...
x LINEAR STRUCTURES
x STRUCTURAL ATTRIBUTES
HIERARCHICAL STRUCTURES
VIEWING 3D
x VIEWING 2D
LOCAL VIEWING OPERATIONS (
APPLICATION ATTRIBUTES + OP.

(x) DISPATCHING (WRITE-THROUGH,
ARCHIVING, ID, DOCUMENTATIO
SIMPLE INSTANCIATION
(x) ATTRIBUTE BUNDLING
NEW ATTRIBUTES
(x) NEW PRIMITIVES
(x) FRAME-LEVEL EDITING
QUERIES ON STRUCTURE

APPLICATION MENUS
NODE / DIRECTORY / FRAME (

DVST (STORAGE)
SIMPLE RASTER (PERI, 41..
LINEAR DIS. LIST (4107...)
PLOTTER
ANY HOST +
VAX + MEGATEK
APOLLO + MEGATEK
APOLLO
APOLLO 3D
MAC INTOSH

x GKS SUPPORTED
FONS SUPPORTED
KP SUPPORTED

(* NOT A DIRECT INTERFAC

PAW – TN/01
29 Aug 85

Towards a Physics Analysis Workstation

R.K.Bock, R.Brun, L.Pape, J.-P.Revol

1. Introduction

We discuss in this note the possibility of development of software for a physicist workstation for data analysis. Such a station would make use of the modern workstation hardware now becoming popular in the High Energy Physics community, and would be based as much as possible on existing software. We think of packages adaptable to different hardware, all based on the idea of iterative interactive creation, hierarchical storage and comparative analysis of statistical information, with possibilities for high quality output. We believe an investment of several man years can produce general tools with considerable impact on that part of physics analysis which uses less CP time than event processing, but takes substantially more other resources on computer systems (tapes, disks, MSS, output devices) and a much larger fraction of physicists' time. The time scale for development should easily meet requirements of LEP experiments.

2. A Physicist's Overview

At present, standard event analysis for most groups ends with the creation of DST-s (data summary tapes). From there on, physicists resort to packages that allow the creation of histograms usually identified by single integer numbers, and their sequential storing by the local system's file manager for later access. Some physicists subsequently make use of histogram manipulation packages (e.g. on VAX systems), but even then the most common output of histograms is on the line printer or consists of inadequate hardcopies from passive terminals. For informal presentations, copies of line printer output with hand annotations are commonplace. Publishable output is hand produced, typically in a drawing office. There is now no accepted way of keeping large numbers of histograms in a manageable structure, accessible to groups of researchers, with additional semi-automated comment and date fields for ease of documentation.

We are convinced that present-day computer host systems with connected personal workstations call for an adaptation of working style. Workstations have brought graphics interfaces easy to deal with even for the casual user. File store, accessible to central and distributed computers, can be used for accumulating histograms with common access for many physicists, and with proper documentation appended. Workstations with their graphics screen can then be used to choose from the histogram base those combinations that show most clearly the intended physics message, and to settle on their best presentation parameters. Finally the small number of publishable graphs can be edited until true publication quality is obtained. A high quality device, possibly central and network-connected, will be interfaced to the stations. It will allow physicists to produce their plots directly and on their own timescale, without having to wait for an overworked drawing office. As all of these steps can be repeated if properly recorded as macro instructions, changes in input data can smoothly be propagated into the final graphics output with minimal work.

We would like to discuss in the following the characteristics of three software packages which would provide us with the necessary functionality. Our firm intention is to provide them an a rather short timescale, say, of a year, with a simple menu-guided user interface, and implemented in a sufficiently modular form to remain open-ended for future evolution.

– Step 1 is executed by a group of service routines to be called by a FORTRAN program (typically a DST analysis job) and providing facilities to create statistically abstracted data (histograms in 1 or 2 variables, tables). We refer to routines very close to a subset of the existing successful HBOOK package [1], run on a computer centre or local host computer. We suggest HBOOK to be amended by various facilities mostly proposed by its many users, and most prominently by a substantial improvement in the generality of its output streams. Facilities are necessary to structure and tag the created objects to ease later comparative analysis of similar histograms. A simple hierarchy of output and the possibility to accumulate histograms produced in several jobs in a single file will result in a *histogram data base*. This base together with a suitable identification scheme will then give access to the output of many jobs to a group of collaborating physicists, avoiding the usual problems of communicating private file name conventions and of histogram identifier clashes. We would also like to create the possibility of a standard output facility for extremely dense micro-DST-s with limited resolution in variables, for later selection upon passing again in a histogramming step. Such substitutes for DST-s could then be kept locally on disk and would undoubtedly make the workstation more efficient for frequent changes in cuts, selection criteria etc. Note that again the proper structuring, commenting, date stamping of files thus generated is essential for collaborative analysis.

– Step 2 is the heart of our proposal: an interactive histogram manipulation program, to run on a small workstation or on a local host (e.g. VAX even with unintelligent graphics terminals), depending on availability. There must be easy access to the file base generated in step 1, including directory services giving access to information about the origin (date, program version, submitter...) of histograms found in the base. Its objective is the comparative analysis of statistical plots and a choice of presentation. Editing possibilities for the resulting graphics file (adding of text, choice of line style, fonts etc.) should be available in a limited form. Portability of the program and of the resulting graphics files should be considered essential. The package should be driven by commands or by menu with command backup, and must contain possibilities for storing command sequences permanently as macros. Graphics output should be in portable (metafile) form, and ample storage facilities for graphics output should also exist, possibly by shipping files to a host machine with large file store and backup. We also envisage taking into account the needs of experiments who may want to use this package in a real-time environment.

– Step 3 finally is an interactive program for the manipulation of graphical information, implementable only on a workstation with modern graphics facilities (bitmap display). This program allows a user (depending on the invested work) to transform graphical presentations into publishable diagrams. Different fonts, area filling, line style, zooming, moving and duplication of picture parts etc. are the relevant functions in this program. General iterative editing (change of style, addition, deletion of comments or picture parts) must be possible, and high quality hard copy output facilities are essential. Access to graphics files created in step 2 (which may, of course, also run concurrently), and facilities for storage of the resulting graphics output must be available. If menu-driven operations are used, the menu commands could be backed up by a command language, which would allow storing command sequences as macros like in step 2.

It is our intention to base ourselves as much as possible on existing software, and to adhere to existing standards wherever this can be done. Workstations are part of the envisaged system, and due to their different characteristics and fast evolution we propose to work in parallel on the implementation for two different stations, Apollo as an example of the upper end which could well run all analysis steps, and MacIntosh as an example for the low-end station which may have limited capabilities for creating histograms. We will also try to remain compatible with the existing popular VAX environment using dumb graphics terminals under HTV. We should also envisage the possibility of installing a MacApollo system as implemented by Phillips et al. at the University of Michigan.

Future extensions are possible and desirable: Depending on the availability of hardware, storing even large files on workstations or on a local file server could become an interesting option, thus migrating most of the histogram creation step onto the local systems and leaving to the host system only the task of creating DST-s. Colour options depend on future hardware and its price; now colour applications are seriously limited because of the lack of economic hardcopy, even Xerox copy facilities. A final, very attractive extension would concern the passage of plots as metafiles into a text processing system for an integral document writing system.

3. A More Technical View

In step 1, the suggested changes to HBOOK are largely esthetic, even at the trivial level of making HBSTAT(0) the default. Not trivial is the identification of output in a hierarchical naming convention and including comments and date information, which is essential in retrieving and managing histograms systematically in what can be considered a numerical data base. It can be hoped that this does not imply backward incompatibility of existing HBOOK files, as the number of HBOOK users worldwide is very large indeed. Compatibility could be achieved by having the identification enter through separate calls, and recorded as separate records, whose absence could be defaulted. A special effort will be needed to include micro-DST options, where highly selected events are represented by few variables which are stored in compressed form. If properly managed, they will allow successive abstraction and selection of data, and allow keeping semi-raw data in the workstation's environment. Here again we stress the need for attributes allowing high level local file management.

Step 2 is the transformation of the number data base (histograms) coming produced in step 1, into a picture data base (metafiles). A package containing the rudiments of interactive histogram manipulation does exist [2], and we should use its elements as far as possible even if majour revisions may be necessary. Also for the command language processor a working CERN product exists [3] and we may be able to incorporate much of it, although the advantages of menu-driven interaction should be seriously taken into account. Again, good identification possibilities and documentation features for the generated output, comprising pictures, command macros, numerical arrays, and functions, must be available. Graphics output should preferrably be portable to other machines and various devices (GKS metafile?). All of the familiar HBOOK functionality for presentation, including curve fitting, axis transformations, error bars etc. must be available in this program. Some desirable overlap should exist between steps 2 and 3, so that some editing tasks could also operate on hosts without a modern workstation. Thus access to different fonts, area filling, line style etc. may be made available in this step.

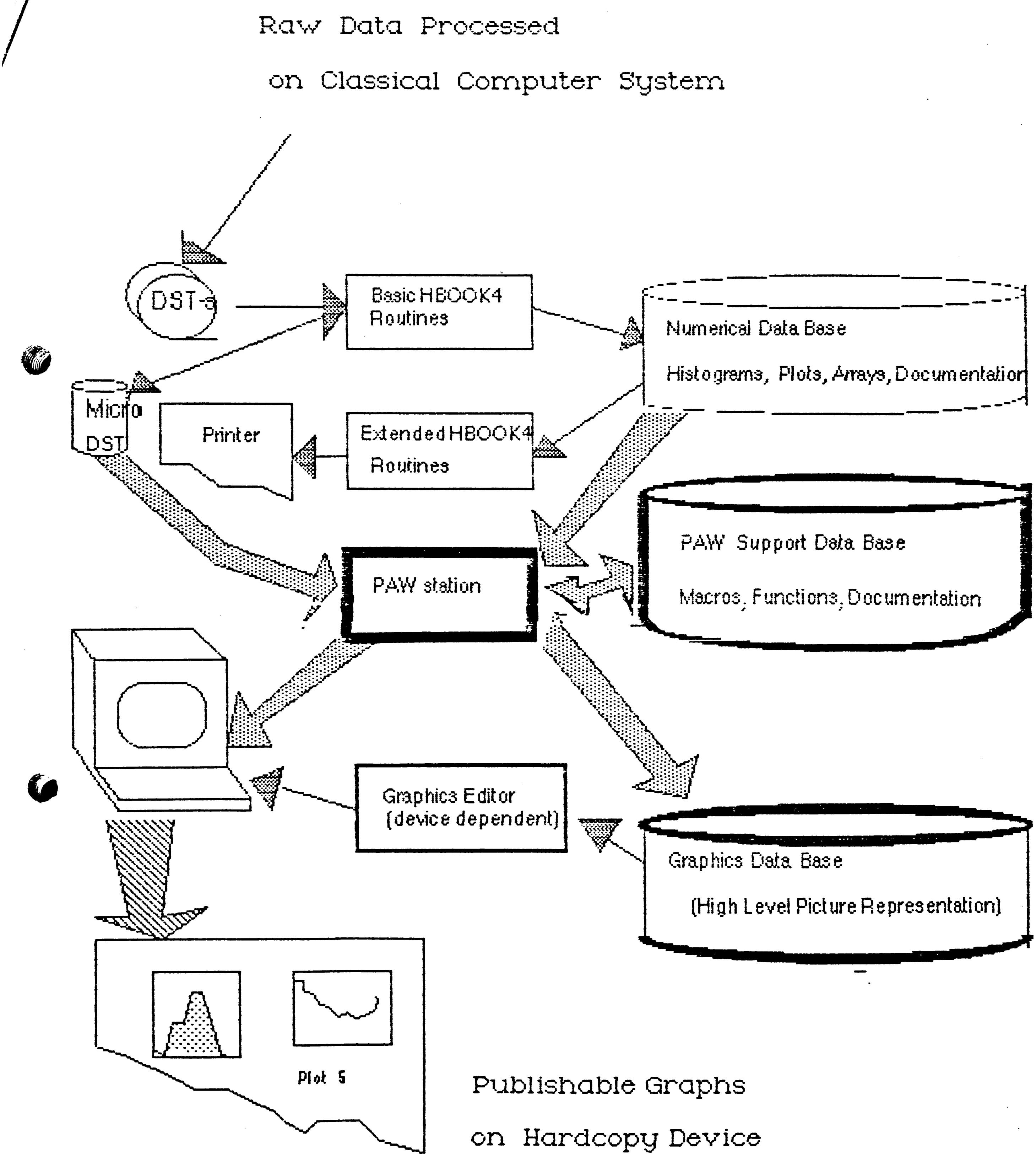
Step 3 is the domain where portability is most difficult to implement. Workstations come with many hardware and software facilities, in rapid evolution, which we must avoid to compete with or reproduce. The proposal to develop in parallel on two systems will give us a chance to learn how to integrate workstations in our daily work, and where the limits of portability are at present. We intend to make use of vendor supplied software like Dialogue (Apollo) and MacDraw or Quickdraw (MacIntosh), but their interfacing will be less than fully machine-independent.

4. An Organisational View

We are discussing a manpower investment that makes sense only if the effort is maintained until completion of a phase 1, to be clearly defined next. The project should therefore be supported by a CERN division, which could be EP and/or DD, and for later maintenance a minimal participation of CERN staff must be postulated. Otherwise, much work can be carried by people on short-term contracts or from outside institutes. There is also some indication that industry might contribute to ease our burden for step 3. This step is of much wider applicability than statistical analysis (in our framework, one could think of event or detector display with graphics editing), and vendors have not failed to recognize the importance of starting such developments.
For input from future customers, an early implantation in running experiments as β-test (serious user feedback) is essential. An obvious point to stress: There should be no preview copy given to people outside the participants, not even for a β-test, before rather complete documentation and help facilities have been made available.

It is our intention to work next on detailed specifications for the three steps, which will then allow to estimate the required manpower and define a phase 1 implementation bounded by available help and a timescale of the order of a calendar year. As from now, we will gladly accept support in specification, design, implementation and testing of the described packages.

System Overview



6. References

[1] HBOOK Users Guide, CERN Program Library, Long Writeup Y250
HBOOK is a FORTRAN callable package for defining and filling histograms and plots from raw data, with a number of presentation and manipulation options.

[2] HPLOT, CERN Program Library, Long Writeup Y251,
HTV , DD report EE/80-5.
HPLOT is the graphics presentation package for HBOOK produced histograms/plots.
HTV is an interactive version of some parts of HBOOK and HPLOT, which allows a user to type in commands and executes HBOOK/HPLOT calls in response.

[3] ZCEDEX, DD report EE/80-6.
ZCEDEX is a command parser for interactive use, and contains features for creating, editing, and storing command macros. It also allows some numerical data manipulation using arrays and user-defined functions.

H.V
20 nov 85

A view on physicist requirements for interactive processing and display

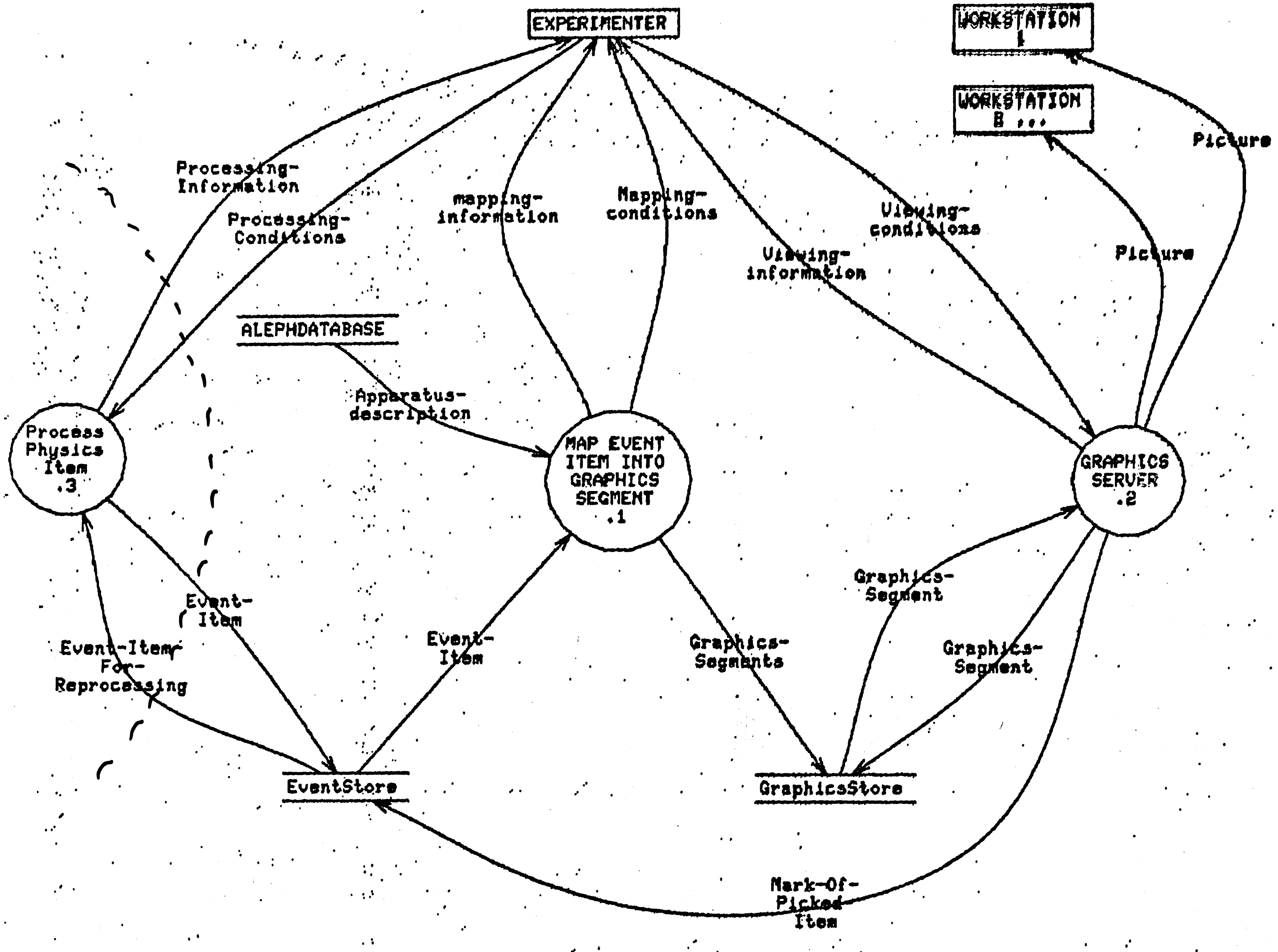
What do you want to do when sitting in front of your screen (s) ?

→ **Run** processes (on data)
((data base))
choose, control, modify, interrupt

Have a look at their results :
event display
(statistics display)

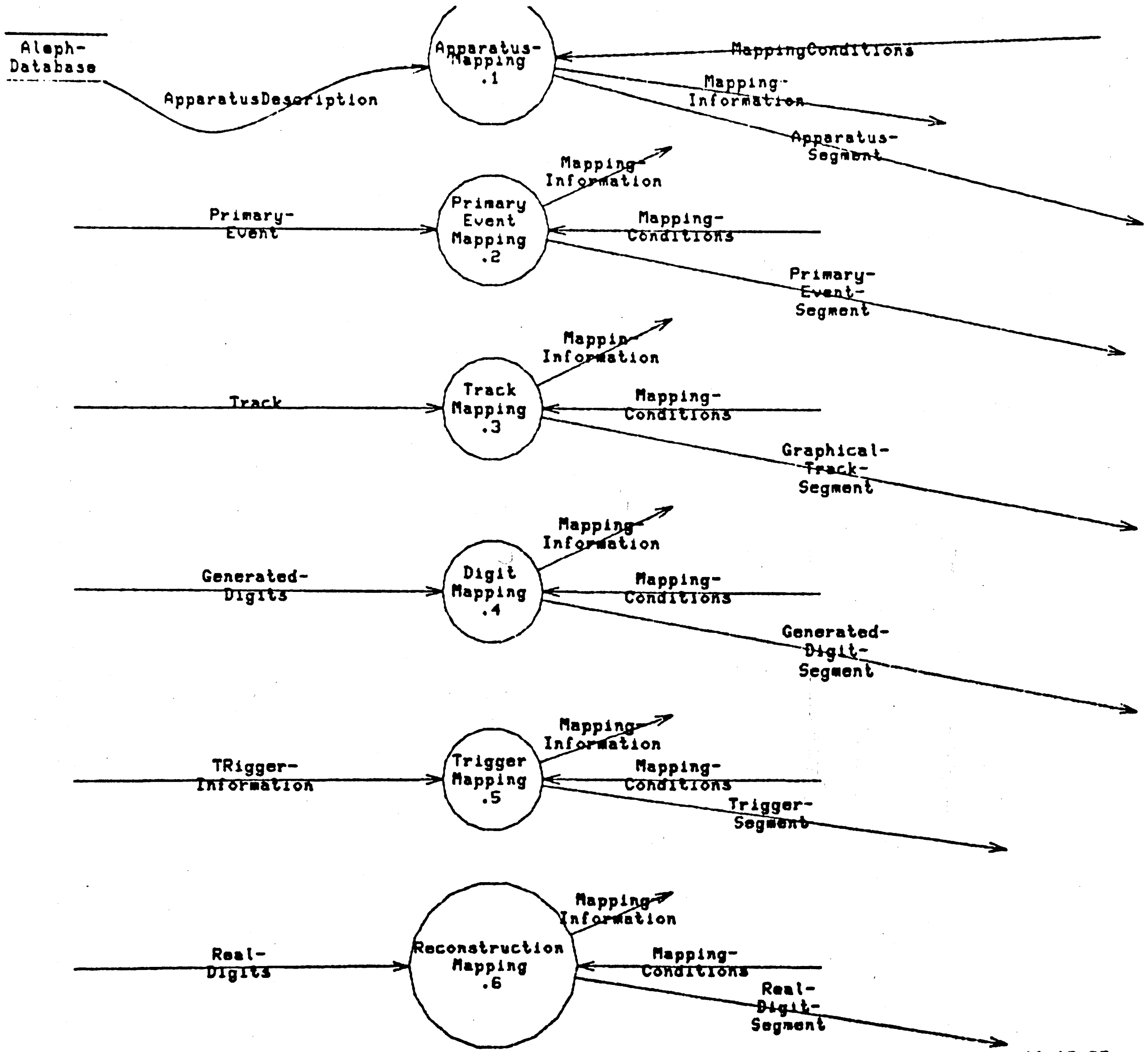
Identify changes to make to | data · process
(keep track for subsequent analysis or replay (journal)

Replay automatically (parts of) the session



11/20/85-videau

DFD 99 - GRAPHICS



11/18/85-videau

DFD 99.1.1- MAP EVENT ITEM INTO GRAPHICS SEGMENT

Implications

Our software has to be made out of:

autonomous modules

- talking to each other via well defined interfaces (stores
- steered at running time by the experimenter or an ersatz of him.
- capable of parallel running to take advantage of modern machine structure
- able of being interrupted

They are basically data driven
They look after their own control
initialisation
conclusion
dialog
interruption

The coherence between modules
does not come

from a rigid, closed program frame

but from

a clear definition of their interfaces

between them (stores)

and to the experimenter

It is insured by the experimenter

with help of menus (propose, not dictat)

automatic pilot (teaching)

by the operating system

Such a structure is open to new packages

it can take advantage of new machine structr

(1 - many)

it tries to protect at best

your efficiency

and your freedom.

Examples exit {off / on}

Development in graphics group

The current machines

VAX's, Apollos (Sun, ...)

provide us with abilities

we need to fulfill our requirements

Let us use them

and not conceive an out of date beast.

working today, nightmare tomorrow.

May I ask

the people currently designing
writing

programs to bear that in mind

and try to structure

their product

a bit that way -

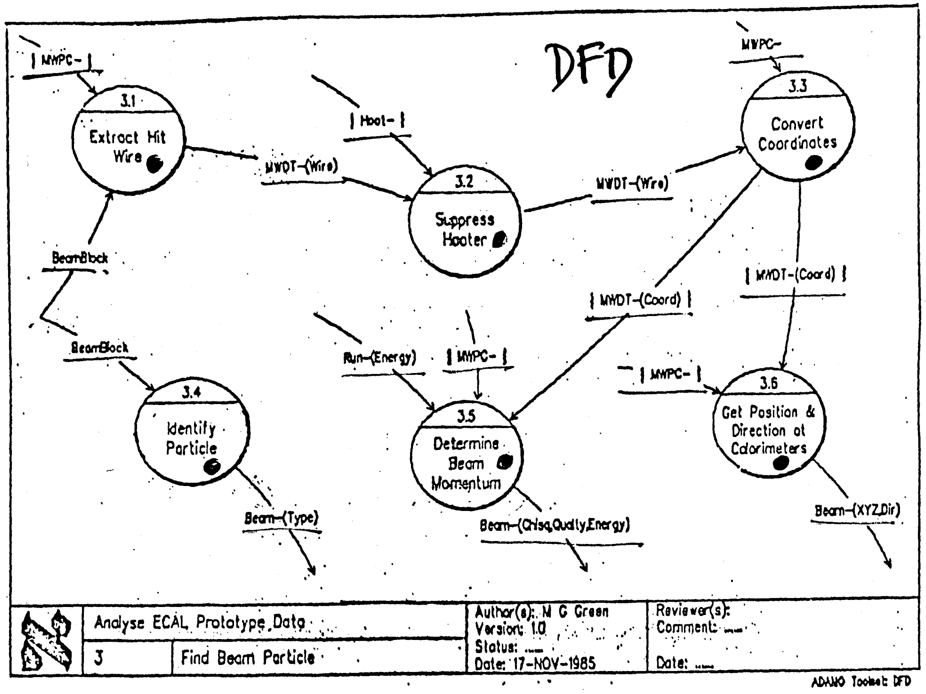
1. ADAMO TOOLS for ER / SA

S.E.

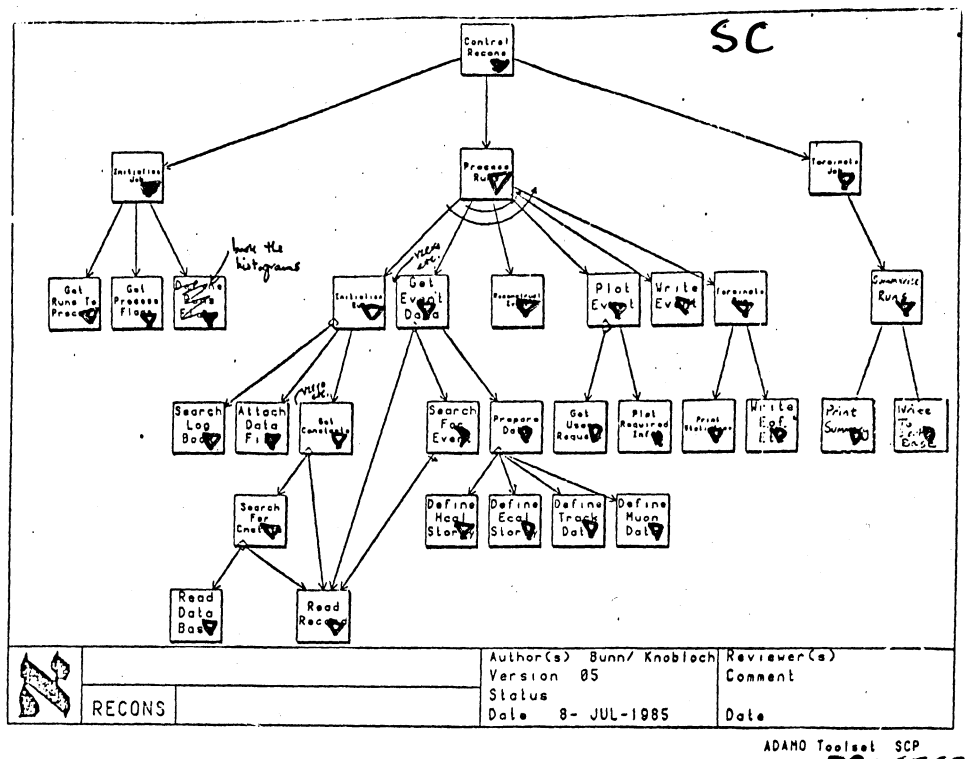
R. BRAZIOLI
S.M. FISHER
P. PALAZZI
W.R. ZHAO

1. COUPLING ER to SA
2. WHAT WE DID
3. HOW WE DID IT
4. WHAT COULD WE DO MORE/BETTER

PROCESS



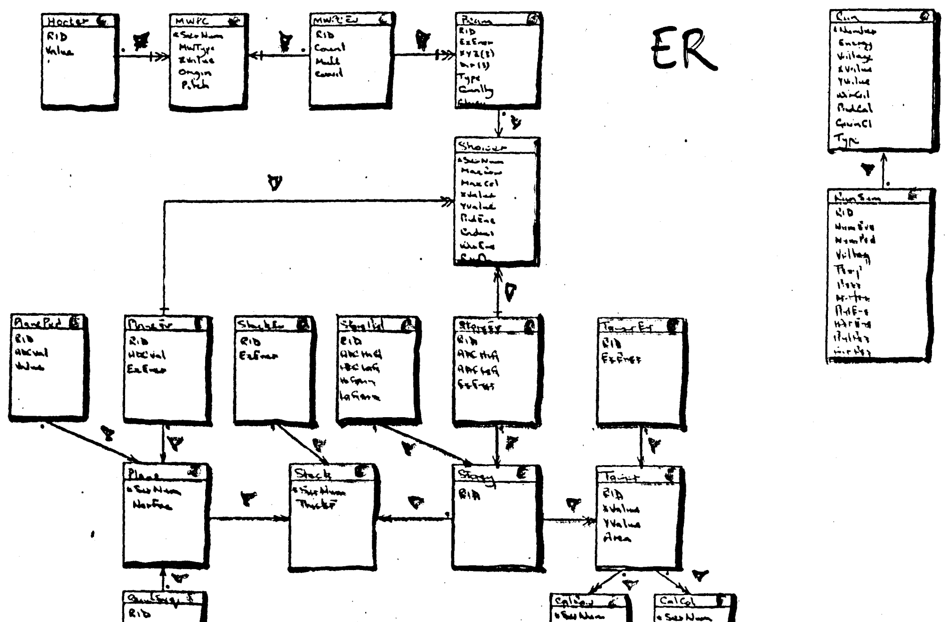
PROCESS



PROCESS

DATA

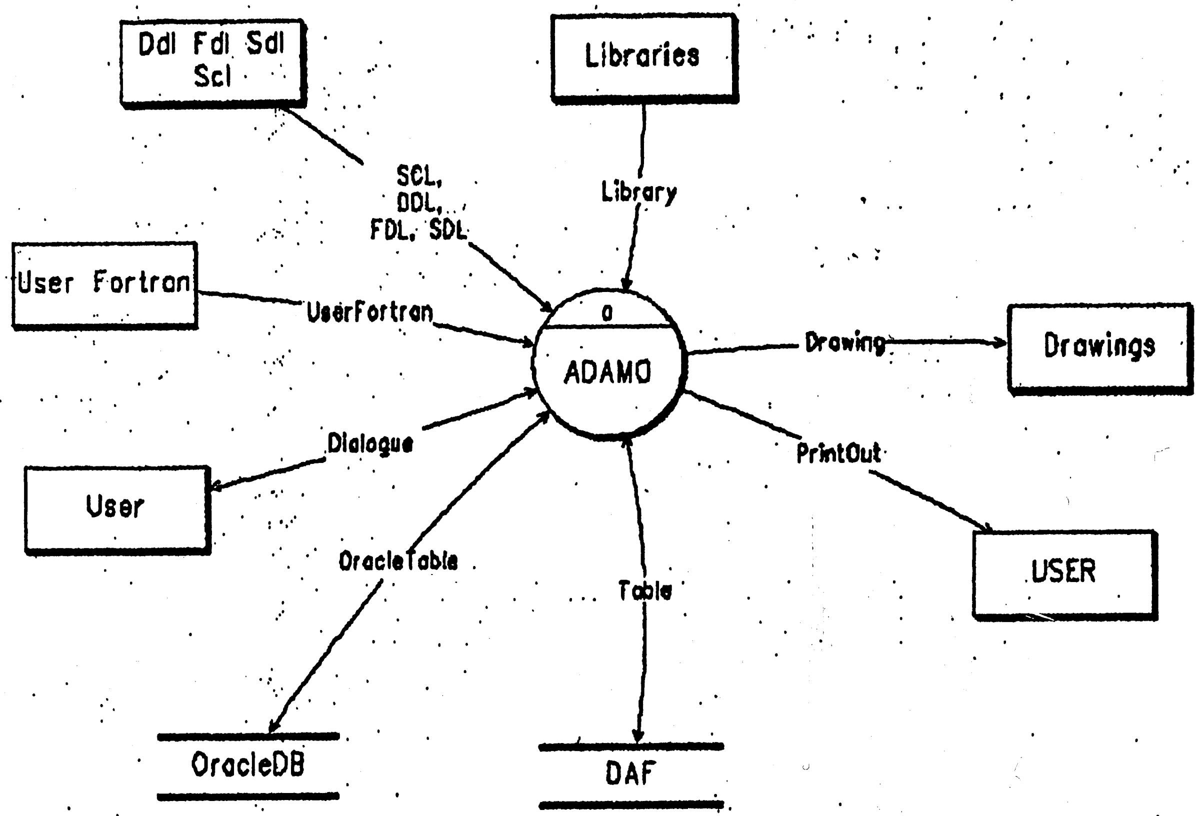
DATA



TEXT WINDOW for DICTIONARIES etc.

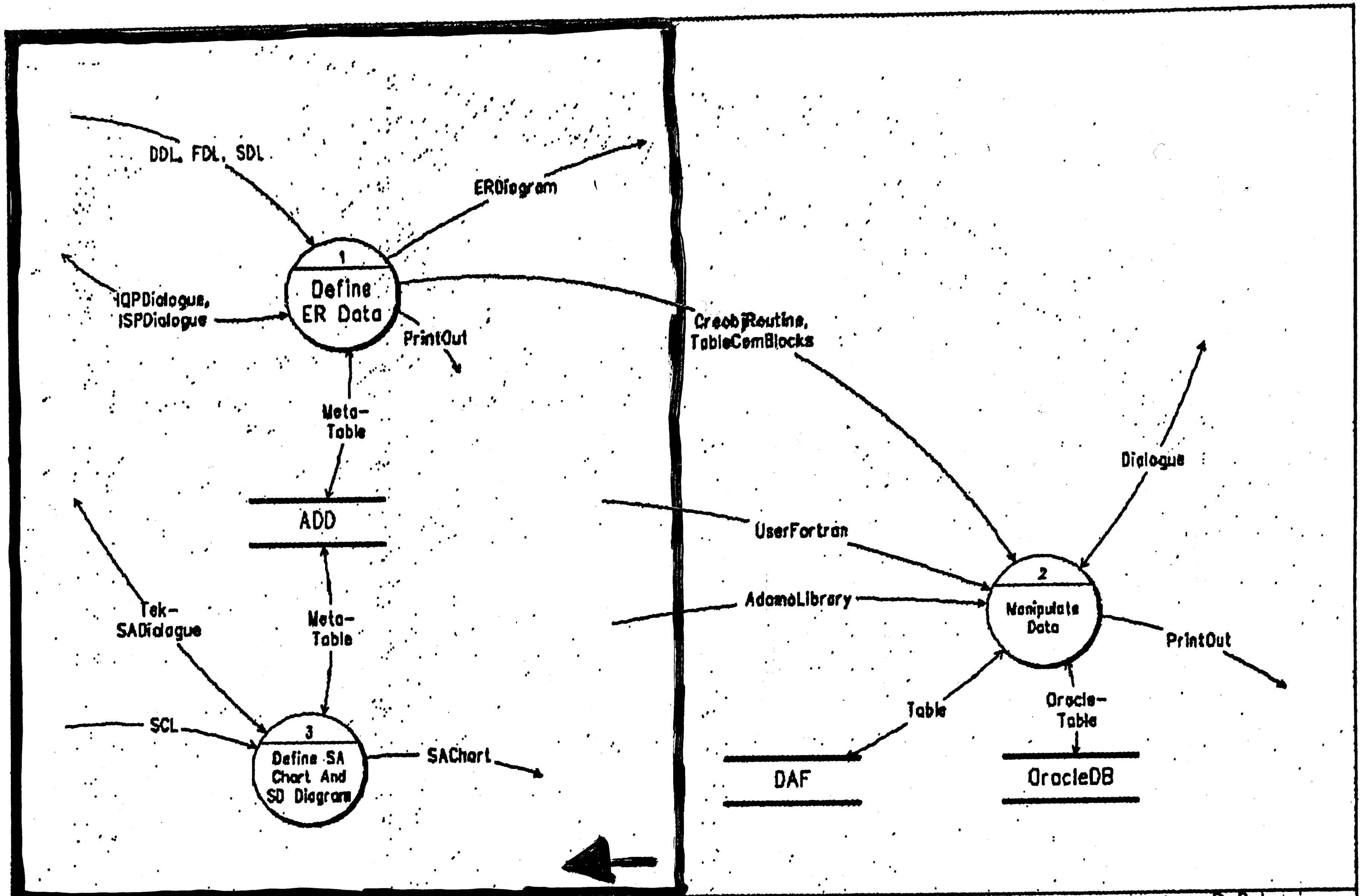
```
SUBSCHEMA Run

DEFINE ATTRIBUTE

NumEve = Number              ;'Number of events analysed';
NumPed = Number              ;'Number of pedestals analysed';
ExEner = REAL                ;'Energy in GeV (but may be just -ve)';
PadEne = ExEner              ;'Shower energy from tower data';
WirEne = ExEner              ;'Shower energy from plane data';
PadRes = REAL [0,*]          ;'Rad energy resolution';
WirRes = REAL [0,*]          ;'Wire energy resolution';
ADCVal = INTE [0,4095]       ;'An ADC value';
ADCHiG = ADCVal              ;'Storey ADC value - high gain';
ADCLoG = ADCVal              ;'Storey ADC value - low gain';
HiGain = REAL                ;'Storey pedestal - high gain';
LoGain = REAL                ;'Storey pedestal -  low gain';
MaxRow = INTE [1,8]          ;'Row with maximum pulse height';
MaxCol = INTE [1,8]          ;'Col with maximum pulse height';
Radius = REAL [0,*]          ;'Shower radius';
BegPla = INTE [0,45]         ;'Starting plane of the shower';
XYZ(3) = REAL                ;'Space point';
Dir(3) = REAL [0,1.000]      ;'Direction cosines';
Qualty = BITP                ;'Beam quality flags';
Count  = INTE [0,*]          ;'Beam MWPC wire hit';
Mult   = INTE [1,*]          ;'Multiplicity';
Coord  = REAL                ;'Beam MWPC coordinate derived from wire hit';

DEFINE ESET

StorePed  =   (ADCHiG,ADCLoG,HiGain,LoGain)
          ;   'Storey pedestal'
          KEY/StoreyPed  (->Storey);

PlanePed  =   (ADCVal,Value)
          ;   'Plane pedestal'
          KEY/PlanePed   (->Plane);

PlaneEv   =   (ADCVal,ExEner)
          ;   'Event dependent plane quantities'
          KEY/PlaneValues (->Plane);

StoreEv   =   (ADCHiG,ADCLoG,ExEner)
          ;   'Event dependent storey quantities'
          KEY/StoreyValues (->Storey);

StackEv   =   (ExEner)
          ;   'Event dependent stack quantities'
          KEY/StackValues  (->Stack);

TowerEv   =   (ExEner)
          ;   'Tower event quantities'
          KEY/TowerValues  (->Tower);
```

2: WHAT WE DID



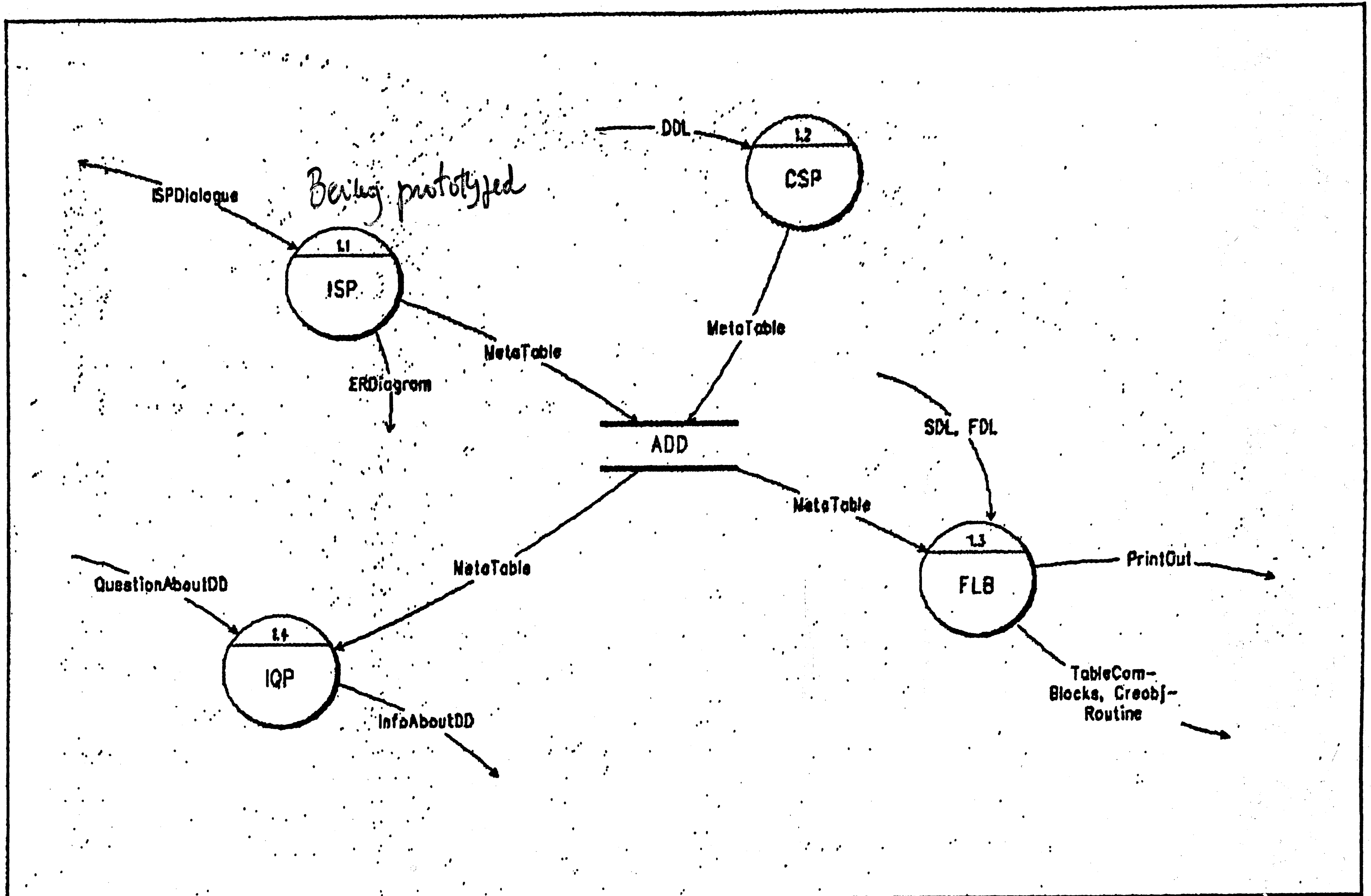
	ADAMO	Author: S Fisher, W Zhao Version: 1.0 Status: ...... Date: 18-NOV-1985	Reviewer(s): P Palazzi Comment: ...... Date: ......
CONTEXT	ADAMO		

ADAMO Toolset: DFD



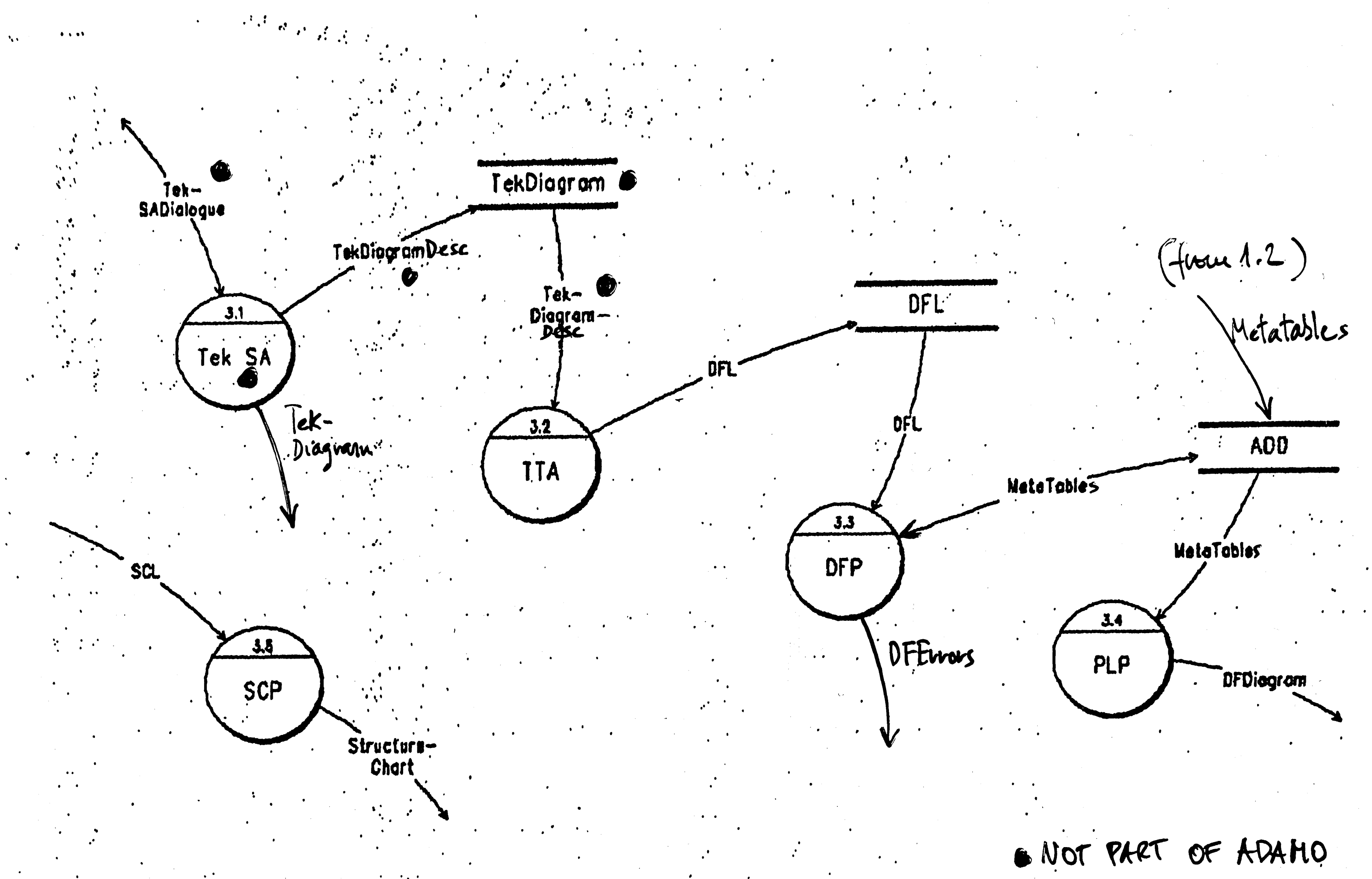
	ADAMO	Author: S Fisher, W Zhao Version: 1.0 Status: ...... Date: 18-NOV-1985	Reviewer(s): P Palazzi Comment: ...... Date: ......
0	ADAMO		

ADAMO Toolset: DFD



ADAMO		Author: S Fisher, W Zhao Version: 1.0 Status: ...... Date: 18-NOV-1985	Reviewer(s): P Palazzi Comment: ...... Date: ......
1	Define ER Data		

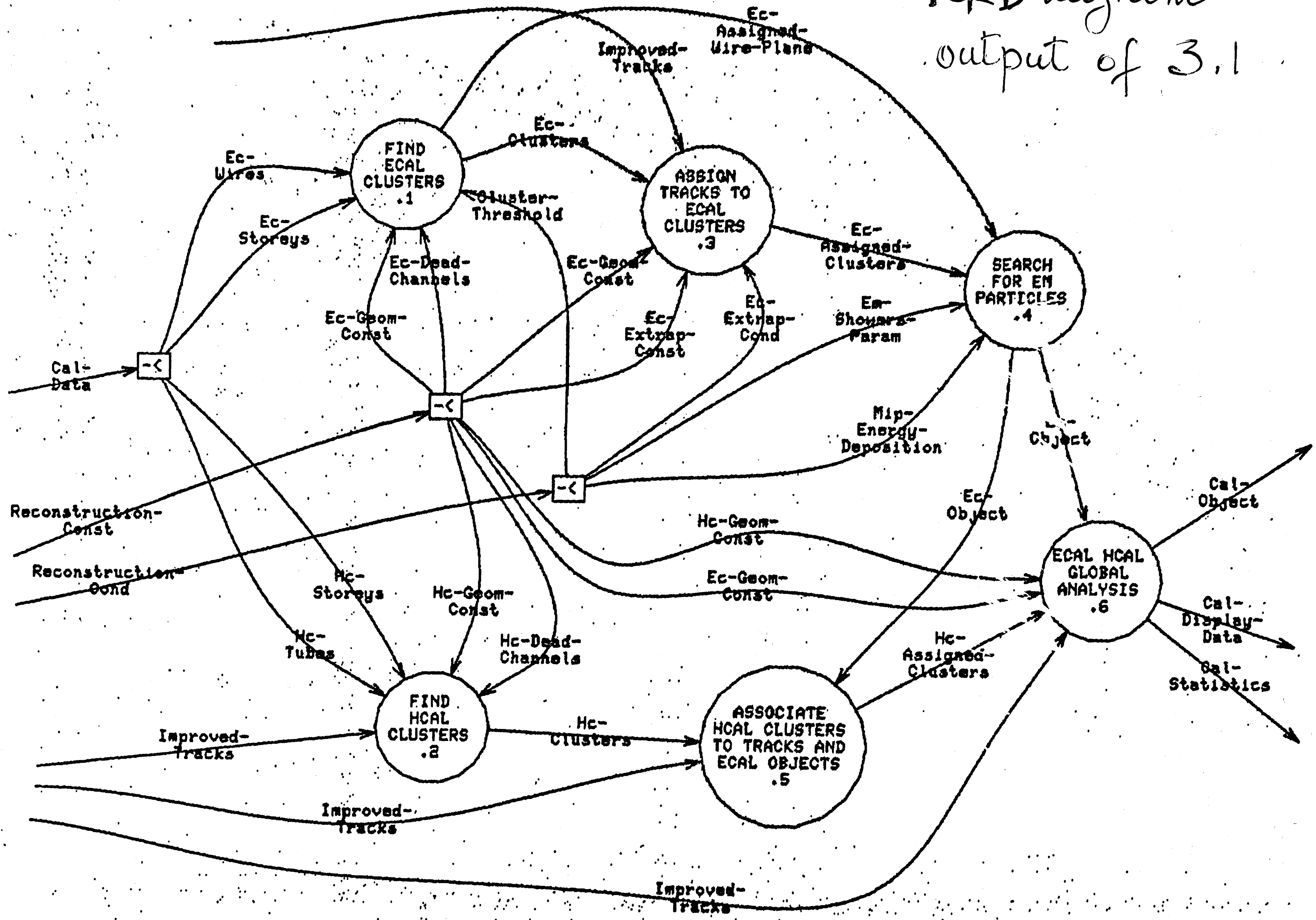
ADAMO Toolset: DFD



ADAMO		Author: S Fisher, W Zhao Version: 1.0 Status: ...... Date: 18-NOV-1985	Reviewer(s): P Palazzi Comment: ...... Date: ......
3	Define SA Chart And SD Diagram		

ADAMO Toolset: DFD

TekDiagram
output of 3.1



11/20/85-knobloch

DFD 2.5 - FIND CALORIMETER OBJECT

Tek DiagramDesc
output of 3.1
input of 3.2

```
RECONSTRUCT EVENTS
knobloch
11/ 7/85
2
DF=
Reconstruction-Const
19     2    -1,-1     -1,-1     -1,-1     -1,-1     -1,-1     -1,-1     -1,-1
DF=
Reconstruction-Const
11     5    516,356     676,336     1644,332     -1,-1     -1,-1     -1,-1
DF=
Display-Data
1     12    -1,-1     -1,-1     -1,-1     -1,-1     -1,-1     -1,-1
DF=
.
.
.
.
11     248,440     3
DI=
CalObjectStore
27     3536,628     1     0
DI=
CalDataStore
16     2280,3084     1     0
DI=
DisplayDataStore
12     888,972     1     0
DI=
MuChDataStore
9     992,2492     1     0
DI=
TrackChDataStore
10     1548,2180     1     0
DI=
CalDataStore
3     1424,1208     1     0
DI=
PREPARE DATA
1     936,1684     0     1
DI=
FIND CENTRAL DETECTOR TRACKS
2     2024,1628     0     2
.
.
.

22     260,656     3
DI=

23     132,2692     3
DI=

28     4068,1620     3
```

DFL
output of 3.2
input of 3.3

```
DIAGRAM '2.5'

AUTHOR    = 'knobloch'
REVIEWER = ' '
VERSION   = '1.0'
SCALE     = (0,150,0,100)
BOXSIZE   = 5
TEXTSIZE = 17

PROCESS

2 : 'FIND HCAL CLUSTERS' AT (51,23);
3 : 'ASSIGN TRACKS TO ECAL CLUSTERS' AT (80,65);
1 : 'FIND ECAL CLUSTERS' AT (47,74);
4 : 'SEARCH FOR EM PARTICLES' AT (115,64);
5 : 'ASSOCIATE HCAL CLUSTERS TO TRACKS AND ECAL OBJECTS' AT (89,21);
6 : 'ECAL HCAL GLOBAL ANALYSIS' AT (123,37);

STORE


DRAW FLOW

HcTubes,HcStoreys
     FROM (  21,   52)                              SAG  -3
    TO 2                                          ;
EcWires,EcStoreys
     FROM (  20,   60)                              SAG   4
    TO 1                                          ;
.
.
.
EcGeomConst,EcExtrapConst
     FROM ( 52,65)                                SAG   1
    TO 3                                          ;
HcGeomConst,HcDeadChannels
     FROM (  45,50)                                SAG   -2
    TO 2                                          ;
EmShowersParam,MipEnergyDeposition
     FROM (  85,55)                                SAG   1
    TO 4                                          ;
ClusterThreshold
     FROM (  15,75)                                SAG  3
    TO 1                                          ;
EcExtrapCond
     FROM (  69,   48)                              SAG  3
    TO 3                                          ;
EcGeomConst,HcGeomConst
     FROM ( 125,10)                                SAG  -3
    TO 6                                          ;

END '2.5'
```

DF Errors
output of 3.3

```
Starting DFP
 ADD input processed                        CPU taken:       6.4600 sec
Starting DIAGRAM
 PROCESS processed                          CPU taken:       0.0300 sec
DIAGRAM processed
Starting DIAGRAM 0
 PROCESS processed                          CPU taken:       0.0300 sec
DIAGRAM processed
Starting DIAGRAM 2
 PROCESS processed                          CPU taken:       0.0900 sec
 STORE   processed                          CPU taken:       0.4500 sec
 LINE    processed                          CPU taken:       9.5800 sec
DIAGRAM processed
Starting DIAGRAM 2.1
.
.
.
 STORE   processed                          CPU taken:       0.0300 sec
 LINE    processed                          CPU taken:       7.3000 sec
DIAGRAM processed
!
! ADAMO/DFP/VALIDITY CHECK:  DIAGRAM  = 2         RECONSTRUCT EVENTS
!                            DATAFLOW = ReconstructionConst
!                            SUB_FLOW = ReconstructionConst
! -------------------------->Missing flow in upper diagram
.
.
.
! ADAMO/DFP/VALIDITY CHECK:  DIAGRAM  = 2.5       FIND CALORIMETER OBJECT
!                            DATAFLOW = CalStatistics
!                            SUB_FLOW = CalStatistics
! -------------------------->Missing flow in upper diagram
!
! ADAMO/DFP/VALIDITY CHECK:  DIAGRAM  = 2.5       FIND CALORIMETER OBJECT
!                            DATAFLOW = ImprovedTracks
!                            SUB_FLOW = ImprovedTracks
! -------------------------->Missing subflow in lower diagram
 VALIDITY_checking processed                CPU taken:      21.4600 sec
Output ADD? (Y/N)
 ADD output processed                       CPU taken:       1.2300 sec
```

From the side → DDL input of 1.2

```
GLOBAL

DEFINE DATAFLOW

/*
    25/02/85, version 6, JMC,IMV,HLV, ALEPH software,
    updated : 15/7/85, GK
    short description :
    this overall diagram shows the interconnection between
    the various major software components and the common
    use of data base information
*/

TOBEDEFINED                                                          ;

PhysicsAnalysisCond
      :'Control of operation modes and modifications to
      parameters and settings needed to run the Physics
      Analysis task'                                                 ;

CalibrationParticles
      :'Input flow to detector which produces known ionization'
          = TpLaserTrack, EcRadioactiveDecayPhotons, TOBEDEFINED     ;


DetectorDescriptionAndParam
      :'Constantants and parameters needed to describe the
      functioning of the detector such as geometrical
      description, alignment measurements, magnetic field
      measurements, results of test beam running with
      prototypes or with final modules, etc.'
          = ApparatusDescription, SurveyConst,
          MagneticField, ElectricField, ExpHallLayout,
          ApparatusBehaviourDescription,
          CalibSourceParam                                           ;

LepMachineData
      :'Information we get from the machine about its running
      conditions, Lep lattice constants, energy, luminosity'
          = LepMachineConst, LepOperationConst                       ;
.
.

SimulationConst
      :'Constants from the DataBase needed by the Simulation process
      during its performance like parameters,settings,..'
          = McInteractionConst, McSimDtConst, McTrackingConst,
          McAnalogConst, McDigitConst, McTrigConst                   ;

EventRecords
      :'Set of SdEvents corresponding to the same trigger'
          = Header, TpEvent, EcEvent, HcEvent, ItEvent,
          TrigEvent, TOBEDEFINED                                     ;

DisplayData
 = MuChDisplayData, TrackChDisplayData, CalDisplayData               ;


Track
 :'Central Detector Track'
  = Id, OneTrFitRes, VtxConFitRes, DedxMeas, ProjCoords              ;

MuChInfo
  = Assign, WtNotMuon, WtDecayMuon, WtPromptMuon,
    IndexMuChInt, IndexMuChExt, IntFlag                              ;

RawEcData
 :'Ecal information issued from the online readout'
 = EcWirePlaneData, CathodeTowerData, TriggerData                    ;

 .
 .
McTrackingCond                                                       ;
LepOperationConst                                                    ;
SurveyConst                                                          ;
EcRadioactiveDecayPhotons                                            ;
TpLaserTrack                                                         ;

   END GLOBAL
```

Metatables
output of 1.2, 3.3
Input of 3.4

TABLE : Bx
COUNT = 13

RID	BxTyp	SeqN	Diag_	Diag_	Diag_	Proce	Sink_	Sourc	Store
1	Process	0	1	75	50	1	=====	=====	=====
2	Process	0	2	75	50	2	=====	=====	=====
3	Process	0	3	75	50	3	=====	=====	=====
4	Process	0	4	20	61	4	=====	=====	=====
5	Process	0	4	127	41	5	=====	=====	=====
6	Process	0	4	121	11	6	=====	=====	=====
7	Process	0	4	58	26	7	=====	=====	=====
8	Process	0	4	96	26	8	=====	=====	=====
9	Process	0	4	120	75	9	=====	=====	=====
10	Process	0	4	62	88	10	=====	=====	=====
11	Process	0	4	77	70	11	=====	=====	=====
12	Process	0	4	47	49	12	=====	=====	=====
13	Store	1	4	138	24	=====	=====	=====	1

TABLE : DesSec
COUNT = 27

RID	Text	DesSe
1	RECONSTRUCT EVEN	2
2	TS	=====
3	RECONSTRUCT EVEN	4
4	TS	=====
5	FIND MUON CANDID	6
6	ATES	=====
7	SELECT INTERESTI	8
8	NG TRACKS	=====
.		
.		
9	COMBINE RESULTS	=====
25	EXTRAPOLATE THRO	26
26	UGH HCAL AND COI	27
27	L	=====

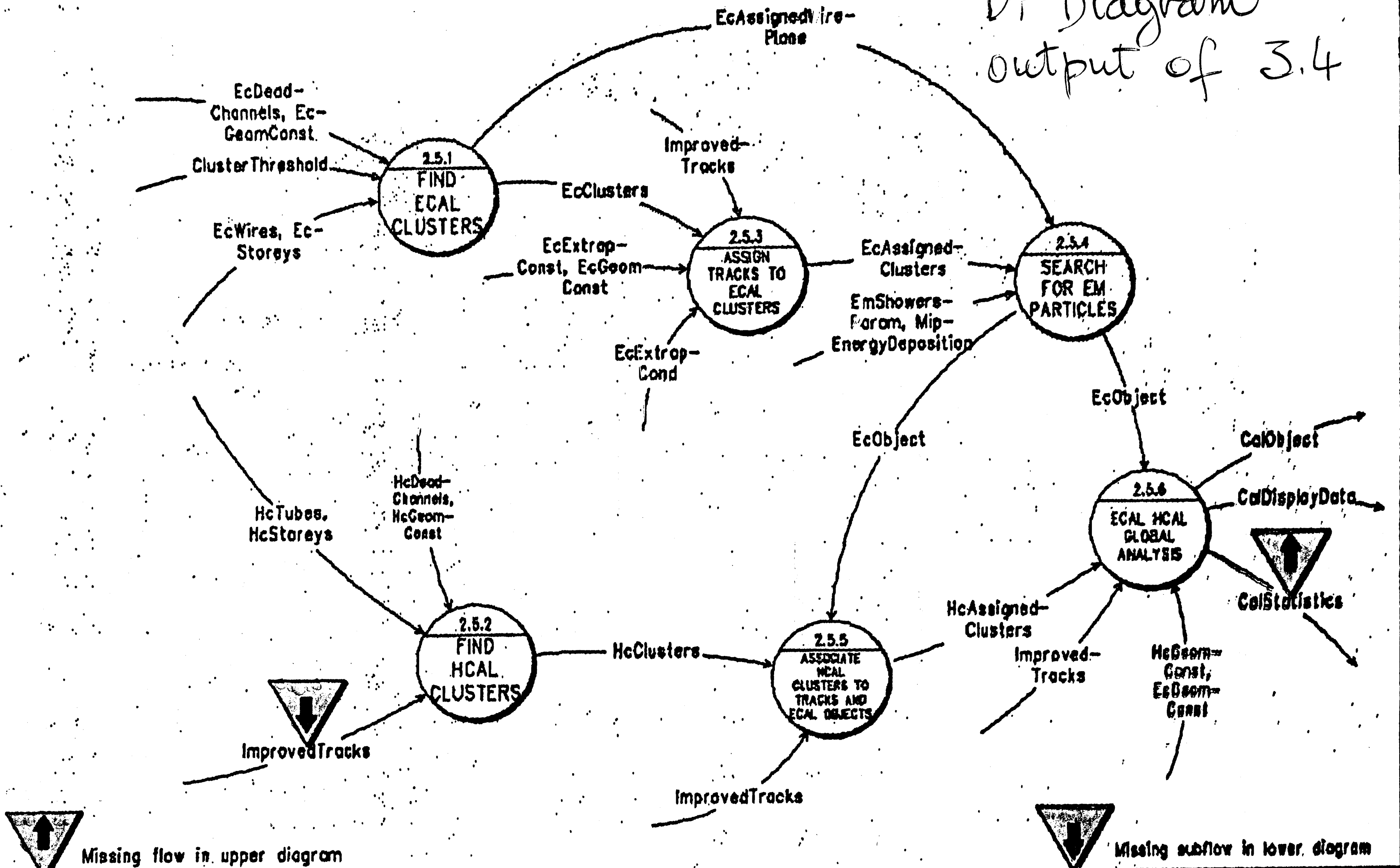
TABLE : LgP
COUNT = 48

RID	LgPTy	Bx_RI	LgP_R	LgP_S	Point
1	Point	=====	=====	0	1
2	Box	8	1	-3	=====
3	Point	=====	=====	0	2
4	Box	11	3	1	=====
5	Point	=====	=====	0	3
7	Point	=====	=====	0	4
.					
.					
45	Point	=====	=====	0	10
46	Box	6	45	4	=====
47	Box	10	=====	0	=====
48	Box	9	47	2	=====

TABLE : Point
COUNT = 10

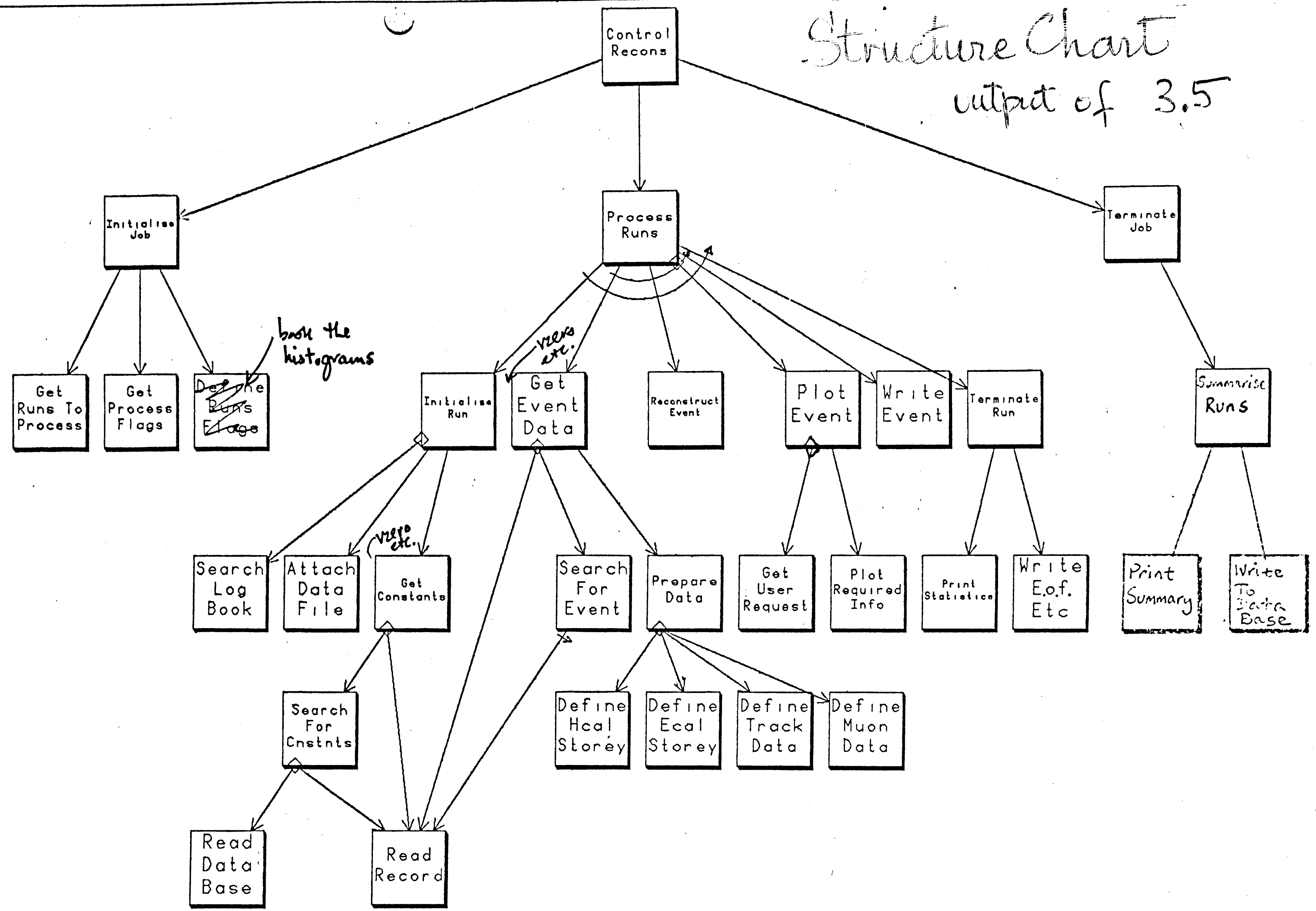
RID	Diag_	Diag_	Diag_
1	4	56	15
2	4	35	70
3	4	38	66
4	4	39	84
5	4	3	49
6	4	148	53
7	4	22	90
8	4	19	37
9	4	56	5
10	4	80	3

DFDiagram output of 3.4



RECONSTRUCT EVENTS		Author: knobloch Version: 1.0 Status: ...... Date: 19-NOV-1985	Reviewer(s): Comment: ...... Date: ......
2.5	FIND CALORIMETER OBJECT		

ADAMO Toolset: DFD



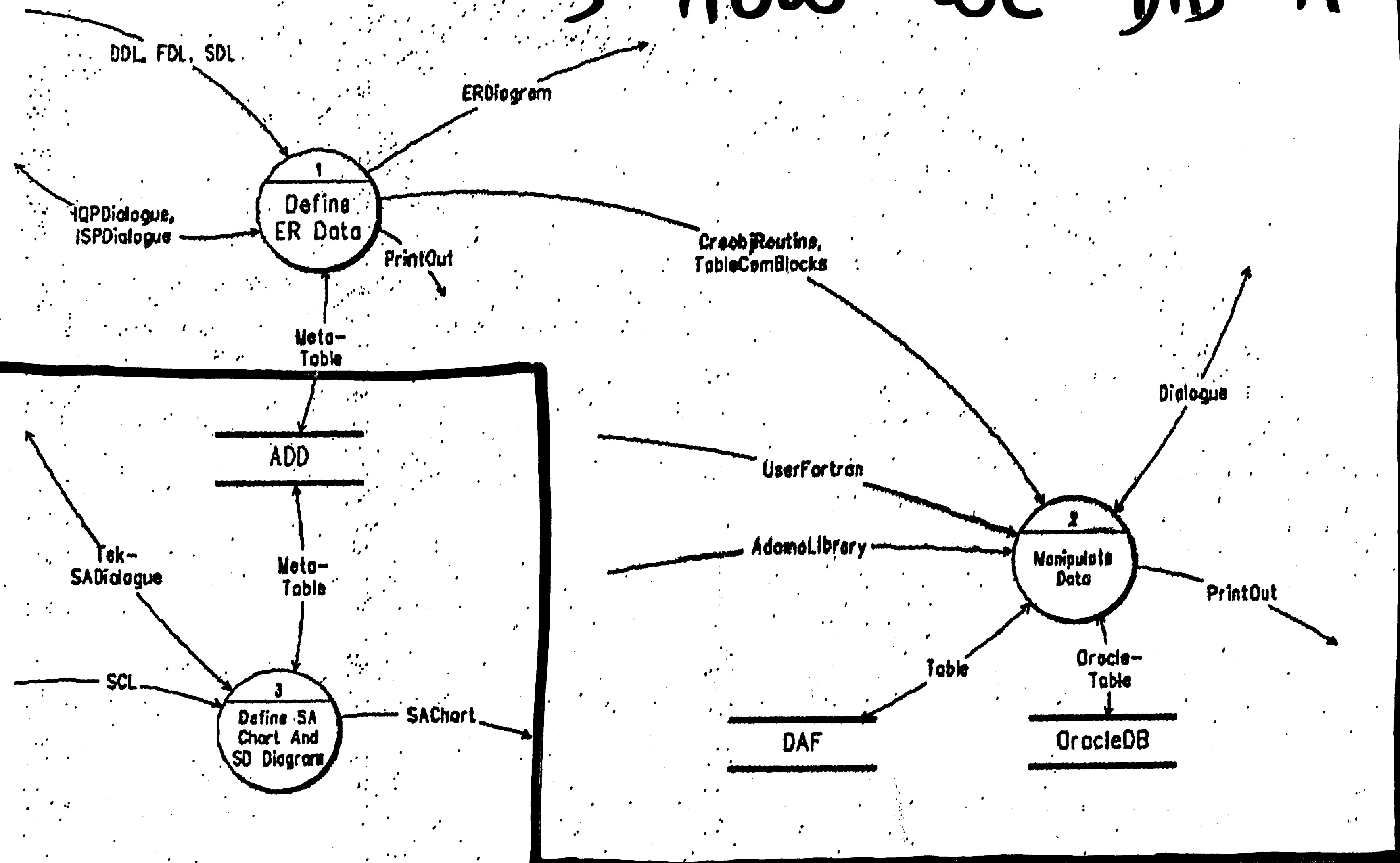
RECONS

Author(s) Bunn/ Knobloch
Version 05
Status
Date 8- JUL-1985

Reviewer(s)
Comment
Date

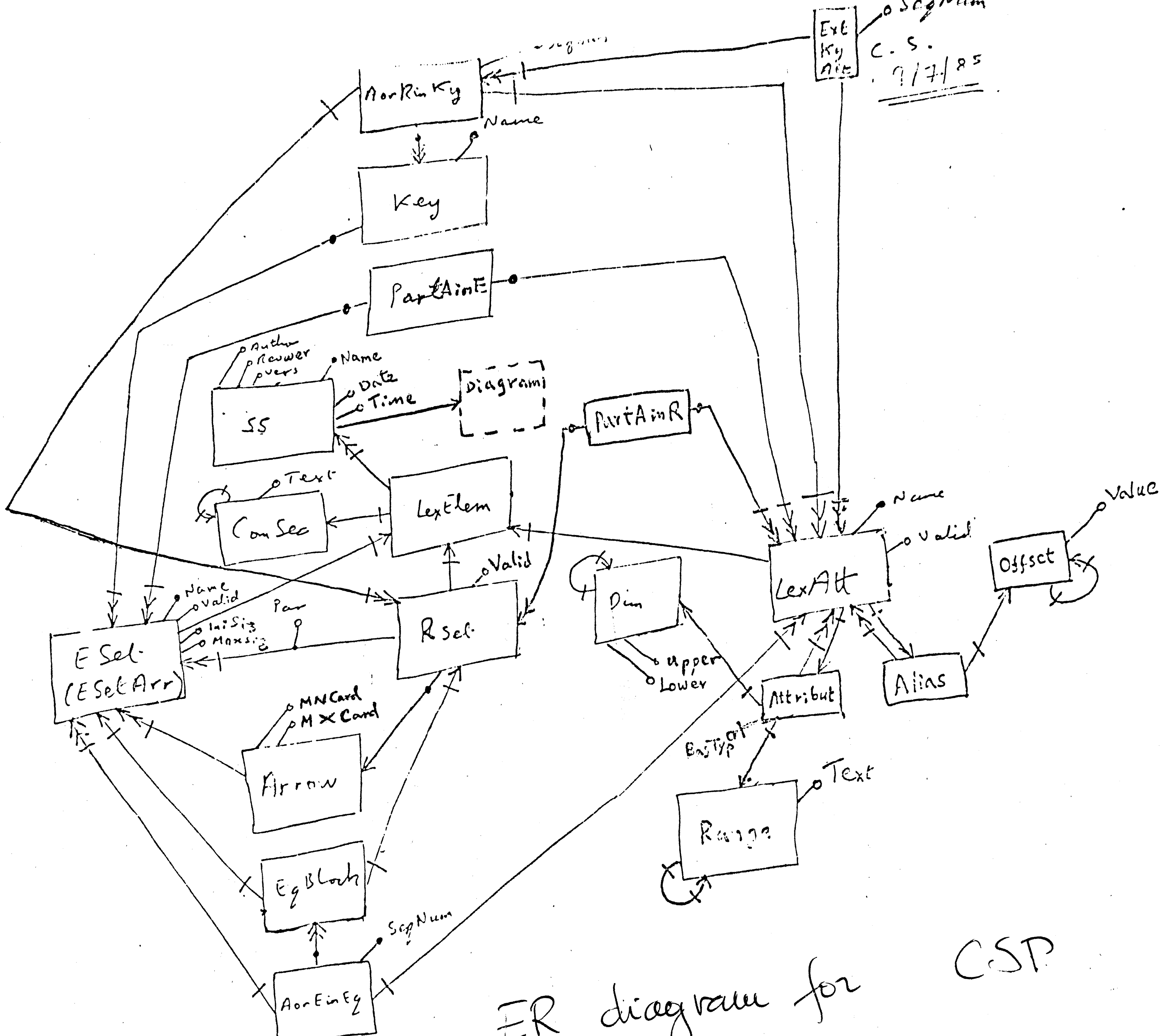
ADAMO Toolset SCP

3. HOW WE DID IT



	ADAMO	Author: S Fisher, W Zhao Version: 1.0 Status: ...... Date: 18-NOV-1985	Reviewer(s): P Palazzi Comment: ...... Date: ......
0	ADAMO		

ADAMO Toolset: DFD

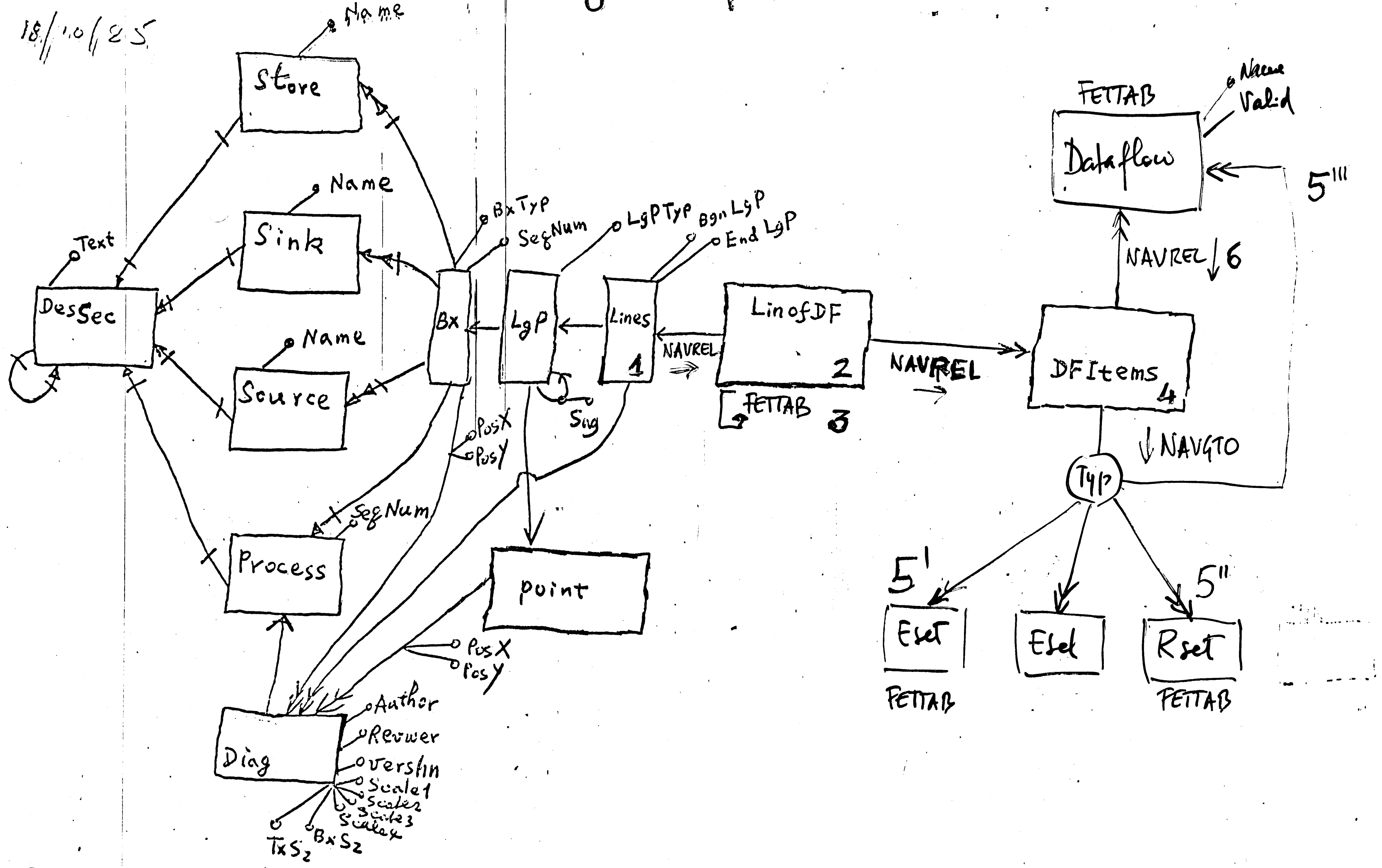


ER diagram for CSP

DFDia

18/10/85

ER diagram for DFD tool



17

A routine of PLP

author : WR ZHAO

```
*DK LTOF
             SUBROUTINE LTOF(SELMD)
             IMPLICIT NONE
C
C  From a line to find its DATFLOWs' name, then all sub_flows,
C  then according to SELMD, put all these dataflows into SEL.
C  SELMD:  'AOfrmFth',  'AOtoFth',  'AOvvFth'
C            'AllfrmSon', 'AlltoSon', 'AllvvSon'
C  SELs are in table DATFLOW, ESET & RSET.
C  Only bottom flows are put into SELs.
C  For father, put both All_SEL & One_SEL.  For son, only All_SEL.
C  For vv, put both frm_SEL & to_SEL
C
*CA HEAD
*CA DFDCM
*CA FLGLBL.
*CA CHGLBL
*CA PARTAP
*CA LINES
*CA LINOFDF
*CA DFITEMS
*CA DATAFLOW
*CA RSET
*CA ESET
C
        INTEGER I,J,CC1,CC2
        LOGICAL OK

        CHARACTER*(*) SELMD
        CHARACTER*16  ENM
C
C  From a line NAV to LinofDF, to DFItems.
C  If bottom, INSSEL according to SELMD. Otherwise recursion.
C
1.2     CALL NAVREL(LINOFDF,FROM,LINES,CCC1,CCC2)
        IF(CCC1.GT.CCC2) RETURN
          DO 80,I=CCC1,CCC2
3         CALL FETTAB(LINOFDF,'Lines_RID',I,OK)
3+4       CALL NAVREL(LINOFDF,TO,DFITEMS,CC1,CC2)
C  If the item is a dataflow?
C
5         CALL NAVGTO(DFITEMS,'Typ',ENM)
          IF(ENM.NE.'ESet') THEN
            ESET_RID=DFITEMS_TYP(2)
5'          CALL FETTAB(ESET,'WINID',1,OK)
            CALL INSMD(SELMD,'ESET')
            GOTO 80
          ELSEIF(ENM.EQ.'RSet') THEN
            RSET_RID=DFITEMS_TYP(2)
5''         CALL FETTAB(RSET,'WINID',1,OK)
            CALL INSMD(SELMD,'RSET')
            GOTO 80
          ELSEIF(ENM.EQ.'DataFlow') THEN
5'''        DATAFLOW_RID=DFITEMS_TYP(2)
            CALL FETTAB(DATAFLOW,'WINID',1,OK)
6           CALL NAVREL(DFITEMS,FROM,DATAFLOW,CC1,CC2)
            IF(CC1.GT.CC2) THEN
              CALL INSMD(SELMD,'DFL')
              GOTO 80
            ELSE
              CALL FRMD1(SELMD)
            ENDIF
          ENDIF
C
 80       CONTINUE
          RETURN
          END

*DK LXCMI
```

for Off-line meeting, November 20, 1985
G.Kellner

Status of evaluation of commercial tools for SASD

- current status of Tektronix SA Tools

  —> installation at CERN
  —> practical experience
  —> problems
  —> what is provided for checking

- other packages (PCSA, proMod, EPOS, Analyst Toolkit, Structured Architect, Argus, . . .)

- Tektronix SD Tools

- home-made tools based on ADAMO (—> Paolo)

- how to proceed from here ?

Current status of Tektronix SA Tools :

Installation at CERN :

- version installed on PRIAM VAX (Unix) since spring 1985 - this works with 40xx and 41xx terminals

- version for VAX/VMS 4.1 has been installed on VXCRNA at the end of September. Same functionality . Graphics editor options and utilities have been improved. **But** - the current version of the graphics editor does not work on 40xx terminals. No information yet from Tektronix if this is a bug or if it is intentional.

- files can be transfered between the VAXes and work continued on either computer, formats are entirely compatible.

Current status of Tektronix SA Tools :

Practical experience :

- graphical input and editing and formal checking of DFD are easy and quite flexible
- all DFD for Monte Carlo, essentially all DFD for Reconstruction and most of DFD for Data Acquisition have been entered on the PRIAM VAX with this tool.
- The DDL has been written with the syntax given in ALEPH note #142 and a conversion program exists which transforms this into the syntax specified by SA Tools.
- Checking facilities of SA Tools have been used to verify DFD and DD. This has been done by converting the DDL to the TEK format, tranfer to PRIAM and run the checking there. One could also transfer the DFD to VXCRNA and run checking there - only the graphics editor does not work properly for 40xx terminals at present.

Current status of Tektronix SA Tools :

Problems encountered :

- **NO** implicit data flow decomposition between levels of DFD. Instead data gates have been introduced by TEK. This is linked to their very simple data dictionary support.

  - **include data gates on DFD to show data decomposition**

    —> good : uses standard features of SA Tools
    DFD and DD can be checked with EVALSA

    —> bad : DFD get very cluttered, sense of data flows gets lost

    ——> examples

  - **proposal by H.Videau to show data decomposition separately**

    —> good : DFD get uncluttered
    data flow decomposition is explicitly visible
    DFD and DD can still be checked with EVALSA (*)

    —> bad : alternating data and process decompositions
    too many extra levels are introduced
    a lot of work to change existing DFD and DD since all ID numbers will change

    ——> examples

- **do not show data decomposition at all**

  —> good : uses standard rules of SASD, **i.e.** datadecomposition between levels

  should be compatible with future upgrades by TEK ?

  existing set of diagrams can be kept, with updates

  —> bad : current version of EVALSA can not be used because too many errors are flagged

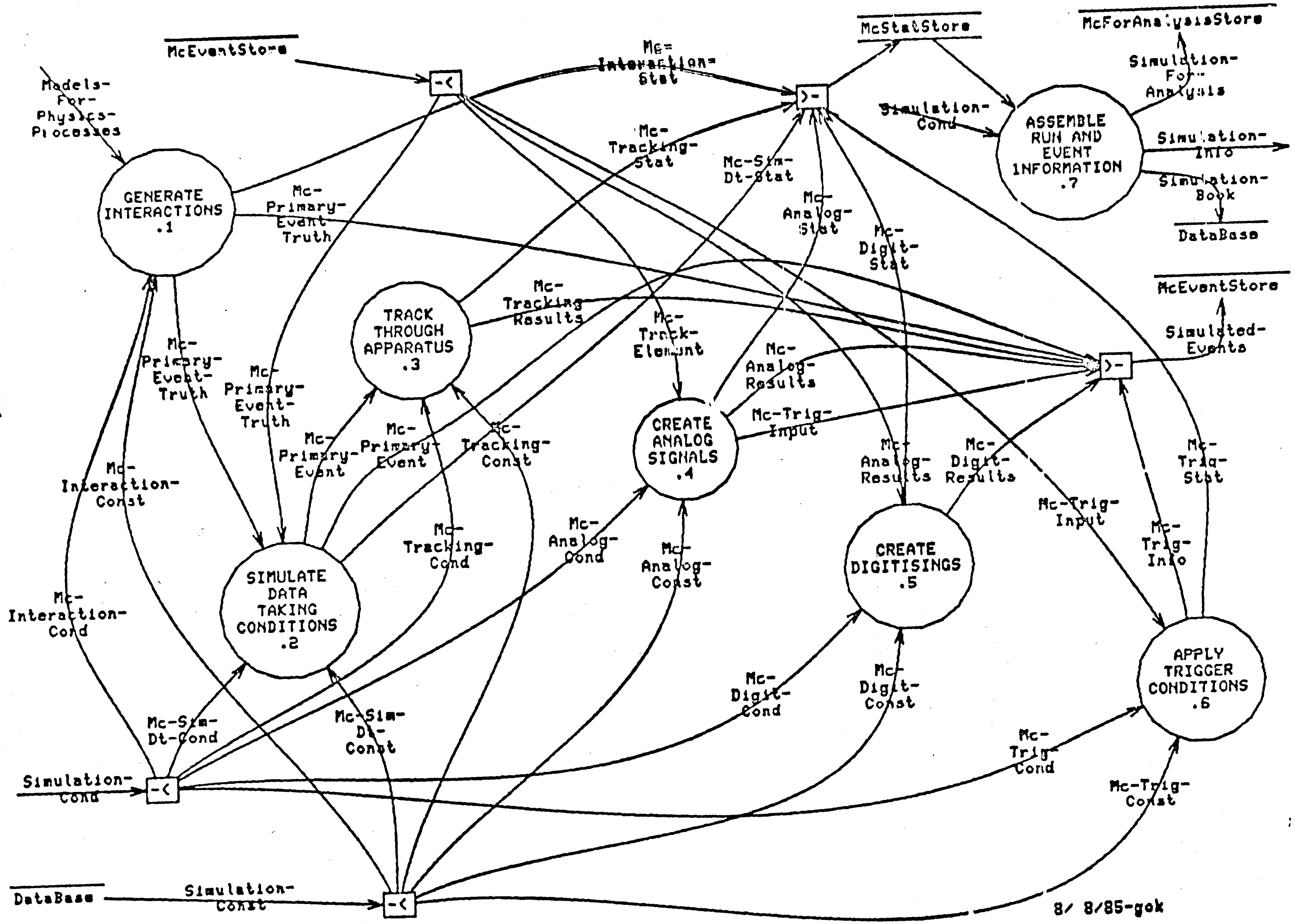
  parent data flow names can not be found at present

  ——> examples

- different **syntax of DDL** for input and TEK Tool is not practical in the long run since several features of checking can not be exploited and logistics is clumsy. Note : our DD format was kept deliberately for a later decision.

••••••••••••> **proposal at the end of the presentation**

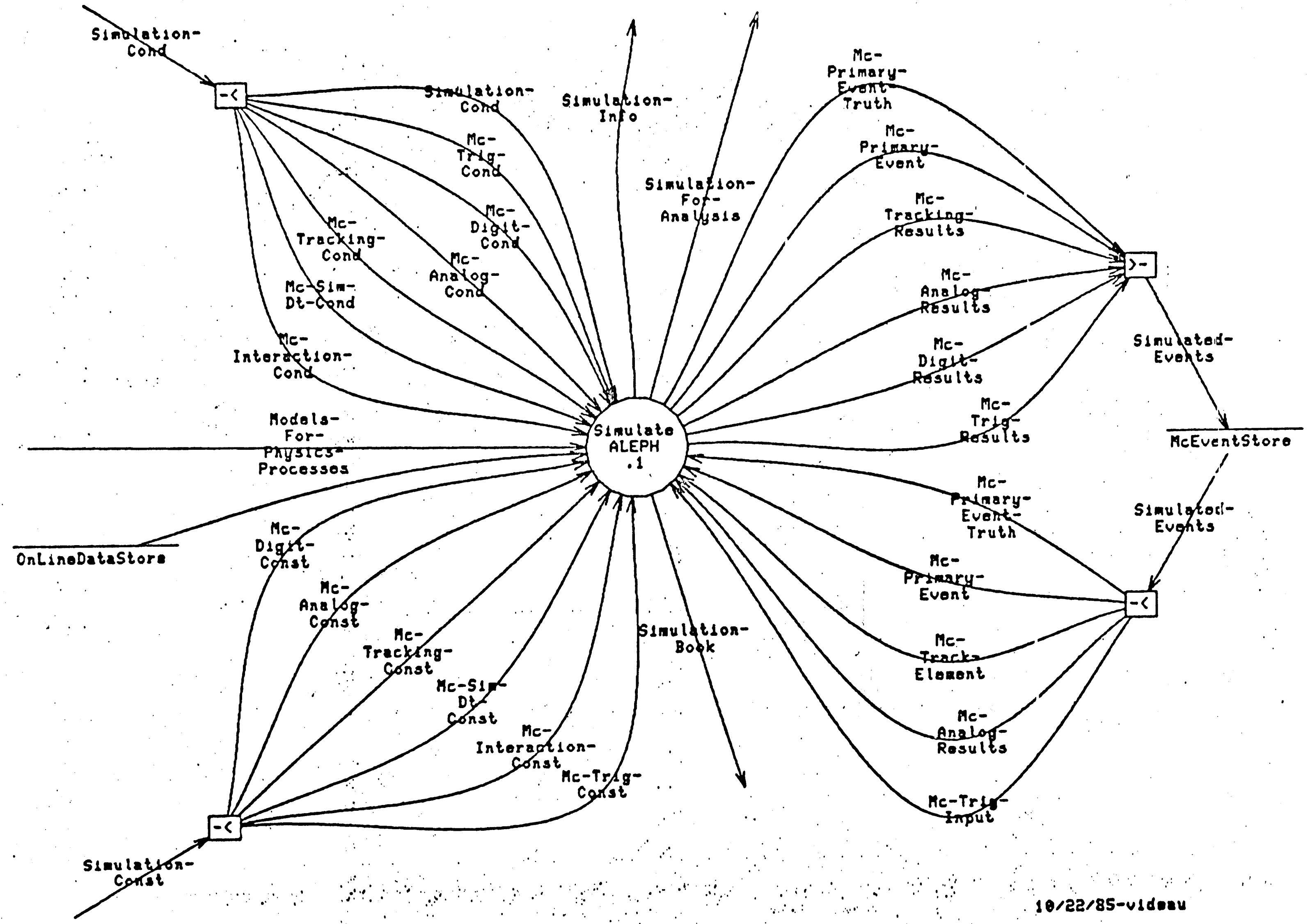
TEK style



DFD 4 - SIMULATE ALEPH

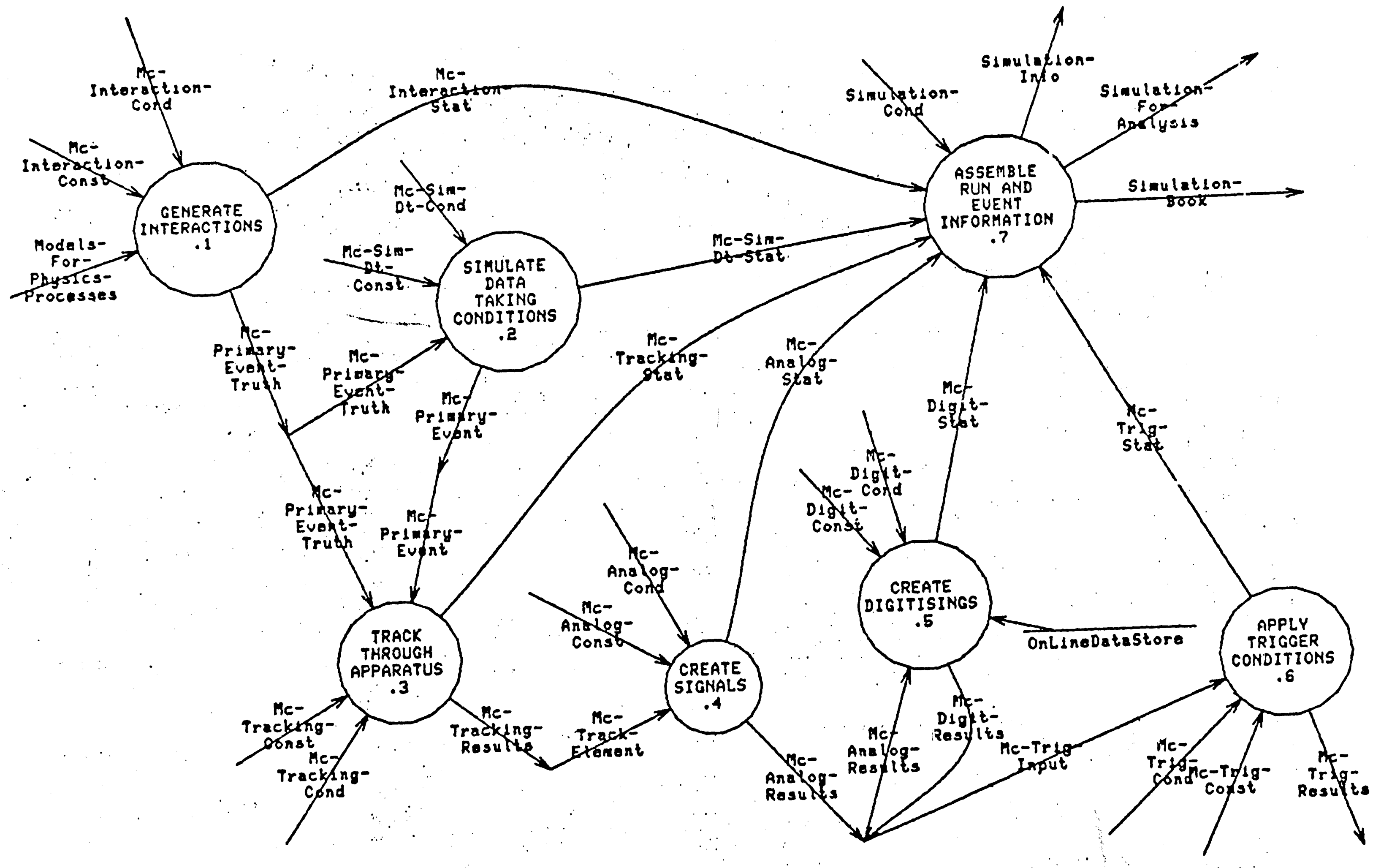
-->pv[5]

H.V.



DFD 4 - SIMULATE ALEPH

H.V.



10/22/85-videau

DFD 4.1 - Simulate ALEPH

Current status of Tektronix SA Tools :

What is provided for checking :

Table 3-3
evalsa Functions

Document	Checked for
Data flow diagram	Valid file type Existing parent Figure number agreement with file name Title agreement with parent Read-only processes or files Write-only processes or files Disjoint data items Number of processes Number of data flows per process Correct file levels Data conservation with parent
Mini-specification	Valid file type Existing parent Figure number agreement with filename Title agreement with parent Length of MS body
Data dictionary	Valid syntax Entries defined more than once TO-BE-DEFINED entries Entries used in DFDs Entries not used in DFDs

Table 3-4
fixsa Functions

Document	Correction
Data flow diagram	Creates the DFD file if it does not exist Changes the title to agree with the parent Changes the figure number to agree with the file name Sets correct file levels Deletes disjoint boundary points
Data dictionary	Creates the DD file if it does not exist Adds those data flow and data store names to the DD that appear in DFDs without corresponding DD entries Removes DD entries that are not referenced in any DFD or DD definition
Mini-specification	Creates the MS file if it does not exist Changes the title to agree with the parent Changes the figure number to agree with the file name

AUXILIARY TOOLS

The tools in this group can be used to perform certain maintenance tasks and to retrieve specific information from your SA documents. These tools are described briefly below, and in more detail in the paragraphs that follow.

addtbd Adds entries to the data dictionary, giving them the definition "TO-BE-DEFINED."

listdf Lists data flow and data store names from the data dictionary or data flow diagrams.

listpnn Lists the process numbers and names from data flow diagrams or mini-specifications.

lookdd Looks up an entry in the data dictionary and optionally displays its definition and the definitions of related entries.

sortdd Sorts data dictionary entries into alphabetical order.

Section 4, Mastering the Tools, suggests ways to use the auxiliary tools with other SA Tools, or with UNIX/TNIX commands, to perform additional tasks.

+ COMBINATION OF THESE + UNIX or VMS ~~tools~~:

- ALL OCCURRENCES OF DF NAME
- ALL DD ENTRIES FOR A GIVEN DFD
- DIVIDE DFD INTO TWO
- AUTOMATICALLY EVALUATE DFD AFTER EDITING
- MODIFY ALL OCCURRENCES OF DF NAME
- UPDATE DD WITH NEW ENTRIES
- etc.

Other packages :

<u>PCSA :</u> used by SPS controls group, very nice user interface, very good response from author(s) for any upgrades requested, very nice features for immediate checking of entries, cheap

BUT - only available for IBM / PC

<u>proMod :</u> has been investigated before SA Tools, very poor user interface, comments by users unfavourable, new version seems to be better, available for IBM / PC and VAX , (too) expensive

<u>EPOS :</u> looks good from the description but not clear about implementation, does not quite follow SASD conventions but can be used for a variety of methodologies

<u>Structured Architect :</u> was evaluated by SPS group and found really inadequate, also available for IBM / PC only

<u>ARGUS :</u> we and SPS group never got more than an advertisement, available for VAX / Unix (VMS ?) and work stations
PS: we just received a video demonstration tape today !

<u>Analyst Toolkit :</u> by Yourdon, particularly suited for SA, with extensions for real-time, at present for IBM / PC and some obscure workstation, no information yet from Yourdon, UK

Tektronix SD Tools :

- being developed, pre-release version should be available by Januray 1986

- discussion with designer and marketing people begin September

- present implementation not too exiting !
  —> DFD can not be transformed to initial Structure Chart
  —> existing DD is essentially ignored
  —> no practical facilities to handle complete SC and select (+edit) and subset starting at given module
  —> etc.

- we do not know when improved version might be available

- doubtful if this would be very useful in the present state

How to proceed from here ?

- investigate how much manpower is required to upgrade ADAMO tools to be used as a full-fledged **SA-SD** support tool set. This concerns especially graphical input and additional features

- take the decision that the currently available ADAMO tools should be used for drawing of DFD and verification of consistency with DD

- continue with Tektronix SA Tools for input of DFD and formal checking until this could be handled by an upgraded ADAMO tool set, provided this is the way we will continue in future

- continue with existing DDL syntax, possibly with some upgrades required for the ADAMO tools

- use the DFD syntax **without data gates** and **without extra diagrams** to show data flow decomposition, **get existing** DFD in this form

- **do not hold back** to update existing DFD and DDL and
  do not hold back to make new DFD, DDL and MS **! ! !**