IMPLEMENTATION OF THE LEP OPTICS MODEL

C. Iselin, J.P. Koutchouk, Y. Marti, J. Poole
CERN
CH-1211 Geneva 23

Abstract To describe the actual LEP accurately, the MAD language has been extended to include, amongst other things, the concepts of a class of elements (e.g orbit correction dipoles) and of attributes of individual elements in the class. It thus becomes straightforward to give a specific name, strength, ..., to any element (like an orbit corrector), a requirement for real accelerators. The LEP descriptions in the MAD8 extended language are stored in an optics database. A LEP run is defined by ~ 10 optical configurations to describe the injection, the acceleration and the β-squeeze. Through a user interface LEPDB, it is possible to access any of the 78 LEP optics, and to link to optics or survey calculations. The optics descriptions, together with the optical parameters (Twiss parameters, dispersions, tunes, ...), are transferred from this database to the new central LEP database under Oracle, on a dedicated computer. A 'run-specification' is added to specify the sequence of intermediate states necessary for a LEP run. A LEP control program, the data loader, accesses the Oracle database remotely and transfers the relevant part. The LEP model has been heavily used during the commissioning and machine studies. It is planned to integrate it soon in the control system proper.

1 INTRODUCTION

The LEP optical model consists of a description of the LEP collider in its various configurations, optics programs and sets of computation results (Twiss parameters, magnet strengths, ...).

It was planned from the design stage that this model should be included in the LEP control system. The motivations have been that the LEP optical configurations are numerous: the insertions may be exploited with/without superconducting quadrupoles, at different tuning levels; the injection is performed after detuning the insertions; the arc cell phase advance is tunable in a large range. The optical programs (control of the tunes, chromaticities, orbits, dispersion, ...) must be adaptable and therefore must fetch optical parameters from files prepared by the LEP optics model. Given the number of possibilities, it was felt that on-line optical calculations would be more economical than the storage of all possible configurations.

Another incentive was to provide a tool that could help understanding the observed behaviour of the machine and would predict on-line the consequences of any modifications to the strengths of magnets and lenses.

2 DESCRIPTION OF THE LEP OPTICS

LEP was largely designed using the MAD program [1,2]. The study of the optics imperfections was carried out using PETROC, a local version of the DESY PETROS program [3], for at that time, MAD did not support imperfect elements.

A different description of the LEP collider was required for each program, very detailed for MAD, simpler for PETROC. They were both limited by the fact that it was not possible, or very difficult to individualise one or a few elements amongst others, e.g. the few orbit correctors needed to perform an orbit bump. This facility is indeed not required at the design stage where all elements of the same type are either identically powered or randomly disturbed.

Modelling the actual collider requires the ability to associate a unique software equivalent with each individually controllable piece of hardware. To extend the standard MAD input language, we borrowed from object-oriented programming in a manner which is illustrated in the following.

1 Let us define a manufacturing line producing a given collider component, e.g. quadrupoles. Each component shares common properties such as the functionality, the length and some distribution of imperfections.

2 Let us further distinguish a subset of these quadrupoles, to be used as skew quadrupoles, i.e. tilted by 45°.

3 Now, let us take one component, give it a name and a position along the reference orbit, and connect it to a power supply.

This description translates into formal language as follows:
let us call each collider component an object.

1a The most important property of the objects is the functionality; it is defined in the standard MAD language : DRIFT, RBEND, QUAD, SEXT, ... They are now called *predefined classes [of objects]*.

1b The manufacturing line produces a set of objects, all in the same *predefined class*, which in addition share other properties. This set is called a *class* and is defined as follows:
<*class name*>: <*predefined class*>, <*parameters*>

e.g.

```
STANDARDQ  : QUAD   ,L=1.6  ! STANDARD QUADs
CV         : VKICK  ,L=0.8  ! V. ORBIT COR.
```

2 Likewise, a subset of a class is a *subclass* of objects sharing more characteristics:
<*subclass*> : <*class name*> , <*parameters*>

e.g.

```
SKEWQ       : STANDARDQ,TILT
```

There is no limit in the hierarchy of classes. When several objects of the same class are connected to the same power supply, they are conveniently grouped into a subclass, where the strength is declared as a standard MAD parameter:

```
QF : STANDARDQ,     K1 = QF.POWER.SUPPLY
```

This parameter is linked to the name of a power converter in the control system database. Setting the QF chain to some value is performed by a standard MAD assignment:

```
QF.POWER.SUPPLY := 0.016013     ! IN 1/M**2
```

3 Each object in a class/subclass is then given a unique name, a position and a strength name if not already done at the class/subclass level. The syntax is as follows:
<*beam line name*>: SEQUENCE
<*object name*>:<*[sub]class name*>,AT = <*position*>,
[<*parameters of the predefined class*>]
ENDSEQUENCE

e.g.

```
LEP      : SEQUENCE
QT1.R    : QTILT  ,AT=  45.673, K1= PSQT1.R1
QT2.R      QTILT  ,AT= 136.793, K1= PSQT2.R1
QD23     : QD     ,AT= 868.943
CV.QD23  : CV     ,AT= 870.647, KICK= PSCV.QD23
                       ...
ENDSEQUENCE
```

Powering e.g. an orbit corrector is simply achieved by the command:

```
PSCV.QD23 := 0.000200 !SET CV.QD23 TO 200 μRAD
```

This new syntax therefore allows one to mimic the hardware set-up. The new version of MAD called MAD8 [2] accepts it as an extension of the standard language.

3 NEW COMMANDS TO COMPUTE OPTICAL QUANTITIES

The MAD program computes tables whose rows refer to individual elements in the LEP machine, and whose columns refer to individual lattice functions and/or element strengths. The LEP control system requires access to these tables by selecting rows and/or columns in an orthogonal way. For this purpose three new commands have been implemented in MAD. The first command, SELECT, marks single or multiple elements in a given sequence for output, e.g.

```
SELECT, SKEWQ  !  Output at skewed quadrupoles
SELECT, CV  !  Output at vertical kickers
```

The second command, OPTICS, selects the corresponding table columns:

```
OPTICS, COLUMNS=S, NAME, BETX, BETY, MUX, MUY
```

This command writes the longitudinal position S, the element class CLASS, and the lattice functions $\beta_x, \beta_y, \mu_x, \mu_y$ in the positions selected by the SELECT command. The OPTICS command writes the requested element parameters for the elements selected by SELECT and the requested lattice functions at the exits of those elements. It is also possible to get at the lattice functions for the centre of each element by adding the flag CENTRE on the OPTICS command:

```
OPTICS, CENTRE, COLUMNS=NAME, S, BETX, BETY
```

An example of such a file is shown in Table 1.

For special purposes a finer control of the longitudinal position is necessary, e.g. for getting the lattice functions in the position of special equipment. The SELECT command can then be replaced by a SPLIT command:

```
SPLIT, B2.QF21, FRACTION=5/6
```

This selects a position at 5/6 of the length of the dipole B2.QF21 for output. In this position there may be, for example, a source of synchrotron light.

4 INTERFACE BETWEEN MAD AND THE LEP CONTROL SYSTEM

Like all programs running in the LEP control system, MAD interfaces with the control system via so-called TFS files [4]. The most-often used files are created by the OPTICS command. MAD also reads the values of, or the increments to the magnet excitations from TFS files.

MAD is also able to perform a closed-orbit correction algorithm using the orbit positions actually measured in the accelerator. A command GETORBIT transfers the monitor readings from a TFS file to the memory pool of MAD. An example is given in Table 2.

The MICADO command then computes the required excitations for the orbit correctors, and the PUTKICK command writes a TFS file containing those excitations which may then be sent to the control system. The actual commands would look thus:

Table 1: An OPTICS Output Table Example

```
@ GAMTR       %f    64.3336
@ ALFA        %f   0.241615E-03
@ XIY         %f   -.455678
@ XIX         %f    2.05279
@ QY          %f   0.250049
@ QX          %f   0.249961
@ CIRCUM      %f    79.0000
@ DELTA       %f   0.000000E+00
@ COMMENT     %20s "DATA FOR TEST CELL"
@ ORIGIN      %24s "MAD 8.01    IBM - VM/CMS"
@ DATE        %08s "19/06/89"
@ TIME        %08s "09.47.40"
* NAME      S            BETX         BETY
$ %16s      %f           %f           %f
  B1        36.6600      24.8427      126.380
  SF1       37.6200      23.8830      130.925
  QF1       39.5000      23.6209      132.268
  B2        75.8000      124.709      25.2153
  SD1       77.1200      130.933      23.8718
  QD1       79.0000      132.277      23.6098
```

Table 2: Example of file with beam position readings

```
@ COMMENT     %20s "DATA FOR TEST CELL"
@ ORIGIN      %24s "MAD 8.15    IBM - VM/CMS"
@ DATE        %08s "10/05/90"
@ TIME        %08s "09.22.51"
* PUNAME    X              Y               STATUS
$ %16s      %f             %f              %hd
  PU.QF1       0.372181E-02  -0.222345E-02     0
  PU.QD1       0.405392E-02  -0.712520E-03     0
```

```
GETORBIT,FILENAME="ORBIT.READINGS"
MICADO,ERROR=1.OE-5,NCORR=100
PUTKICK,FILENAME="KICK.SETTINGS"
```

5 THE LEP OPTICS DATABASE

The descriptions of the LEP optics in the standard and extended MAD languages are maintained in the LEP OPTICS database. It was designed in 1983 for Design purposes mainly. However, it was soon found to be a useful tool to update and transmit the information to other groups (SURVEY, VACUUM, INSTALLATION, PLANNING, MAGNETS) and consequently the database was extended to include vacuum and instrumentation elements. It is installed on VM and managed with the CERN file management system PATCHY.

The database contains the description of the LEP layout in two forms:

1. The LINE form (standard MAD language) which describes the machine at different levels:
   - level 1: contains magnets, drift spaces and interaction points
   - level 2: contains level 1 + monitors, correctors, collimators, RF cavities, electrostatic separators and markers at each half-cell.
   - level 3: contains level 2 + dipoles divided into three pieces and beam instrumentation elements.
   - level 4: contains level 3 + markers for the vacuum chambers.
2. The SEQUENCE form (Extended MAD language) at description level 3 . This has been introduced lately and is derived from the LINE form.

The database also contains all the parameters, magnet excitations and field imperfections, describing each optics (about 140 optics versions up to now).

On VM a tool called LEPDB allows the user to extract information from the database and optionally run MAD via a menu-driven interface. In combining the structure, the related optics parameters, calculation and output commands, the user builds a complete, flexible, modifiable input data for MAD.

This facility is particularly helpful when preparing the information needed to load machine-description tables in the control system ORACLE database.

6 THE ORACLE DATABASE

The structure of the database reflects the main features of an optics: the machine is comprised of components which are connected together and contain magnetic fields. The machine is described in terms of the layout, and the physical properties of the elements. An optics is defined by the strengths and is described by the optical parameters of the beam. These data are stored in 4 tables: geometrical data (element name, class, location, strength name,...), element properties (class, length, tilt etc.), description of the optics (optics ID, β^*, tunes, etc.) and finally the magnetic strengths which will produce the optics (optics ID, strength name, strength value).

It is possible to produce an input for MAD from the Oracle tables and produce the lattice functions ('twiss data') which are required for many of the application programs in the control system. As mentioned above, the converse is also possible, i.e. data can be loaded into Oracle from MAD input. These connections to Oracle use the Oracle networking product and therefore work on all of the host computers running MAD at CERN.

7 THE LEP RUN SPECIFICATIONS

A run in LEP consists of a sequence of pre-defined machine states for which optical configurations are combined with RF configurations. Each state is defined by a unique sequence number, the energy, a transition time from the previous state, a pointer to the optics data, a pointer to the twiss data, a pointer to the RF data and finally a flag indicating the mode (injection, ramping or beta squeezing).

The run specifications are stored in Oracle and transferred to the control system in the normal way (see below).

8 DOWN-LOADING TO THE LEP CONTROL SYSTEM

All reference data are stored in Oracle and are loaded into the control system using the Data Loader. The control system Data Loader comprises tables containing SQL statements, used to gather and extract data for the control system, and software for making the transfer.

Data in the control system are stored in files generically known as the Reference DataSet (RDS) [5]. These files are necessary to allow rapid and selective access to the data by application programs. The problem of using a copy of the source data is solved by a watchdog program [6] which checks the copy against the source and generates an alarm when an error is detected.

The data in the RDS are structured according to the system which they describe e.g. vacuum, power converter, optics etc. There is a one-to-one mapping between the hierarchical directories/files of the RDS and the systems/sub-systems in the loader database.

The optics/model data are to be found in the run specifications system where there are sub-systems for magnetic strengths, optics descriptions and twiss data. Application programs are directed to the relevant data through the run specification and the run table which summarises the actual machine state.

9 CONCLUSION

The MAD extensions and the data-base facilities have proven their usefulness, especially during the commissioning. Drastic and unexpected optics changes were tested and used for LEP very easily. The user interface software layer has only been implemented in a primitive form and will be improved in the light of experience gained during operation.

References

[1] H. Grote and F.C. Iselin, *MAD Version 8, User's Reference Manual*, CERN/SL/90-13(AP), 1990.

[2] H. Grote and F.C. Iselin, *Recent Developments in MAD Version 8*, Paper FR P 48 C at this conference.

[3] J. Kewisch, K. Steffen, The PETROS optics program, DESY PET-76/09 (1976).

[4] Ph. Defert, Ph. Hofmann, and R. Keyser, *The Table File System, the C Interfaces*, LAW Note 9, CERN, 1989.

[5] J.P. Koutchouk and J. Poole, *Data Management for LEP Control*, Proceedings of the IEEE Particle Accelerator Conference, Washington, D.C., March 1987.

[6] J. Poole, *The Database Systems for LEP Control*, Paper WE P 07 R at this conference.