MEGA: MONTE CARLO EVENT GENERATOR ADAPTOR

F. Anselmo, E. Barberio, G. Bruni, P. Bruni, L. Cifarelli, F. Cindolo, O. Di Rosa, G. Iacobucci, G. La Commare, M. Marino, R. Nania and A. Zichichi

Abstract

Different output formats from existing event generation programs for Monte Carlo simulations in High-Energy Physics have been translated into a standard, flexible and transparent data structure via a new software package: the Monte Carlo Event-Generator Adaptor (MEGA). The MEGA data structure is based on the Entity-Relationship (ER) data model of the ALEPH Data Model (ADAMO) system. It is endowed with a set of appropriate interfacing routines, thus making it possible to easily and uniquely connect different event generators with a given tracking and/or analysis programme. An interesting application of MEGA in the context of the GEANT tracking program is presented.

1. Introduction

Future experiments in different kinds of high-energy accelerators will require extensive theoretical and phenomenological studies for extrapolation and/or prediction of possible physics scenarios, as well as detector design studies to define the required construction parameters in terms of dimensions, performances, resolutions, etc. [1]. In very many phases, these studies will be accomplished via the Monte Carlo simulation.

There already exists a large collection of event generation programs, reflecting different theoretical approaches to the description of various physical processes. As a consequence, in order to accomplish the above studies for a given type of interaction, one is often confronted with different event generators (EGs). The information provided by these EGs may differ depending on the underlying physics model and, even for identical quantities, their code and format presentation may widely vary. As a net result, the handling of different EGs often requires some unnecessary waste of time. This is true for any analysis program, in particular when a comparative study of the results from different EGs has to be performed. Moreover, in order to simulate the detector response by using a given tracking program (TP), a considerable amount of time is usually spent developing interface routines whose algorithms have to take into account the diversity of the EG output data structures with respect to the TP input data structure. The situation becomes even more complicated when more than one TP is considered. Some standardization in the organization of these data would certainly result in a much simpler job.

Efforts to solve such a problem have been, for instance, the Particle Data Group suggestion [2, 3] of a standard numbering scheme for particle identification, and recently, the LEP Physics Workshop proposition of a standard common block to store the EG output data [4], the so-called High-Energy Physics EVenT (HEPEVT) common block which has been well accepted and already introduced into some of the major simulation programs (sect. 3).

An alternative solution for the arrangement of a generalized Monte Carlo output is presented herein. The adopted point of view is that nowadays this kind of problem can only be efficiently dealt with using up-to-date instruments like data modelling techniques, which automatically provide a number of advantages in the subsequent phases of data processing [5,6]. In particular, it has already been stressed [7] how complex programs with large databases, as needed in the HEP environment, may considerably gain in simplicity and flexibility when relational models dealing with dynamical memory structures are used.

The solution, which is going to be described in the following, is based on the Entity-Relationship (ER) data model originally proposed by P.P. Chen [8] and implemented in the ADAMO system [9]. According to this model, data are mapped on tables, the so-called "entities" with their "attributes", which are linked by mutual "relationships". In addition to its suitable underlying data structure, the ADAMO system provides a set of powerful table-handling tools [10, 11] which allow the User either to construct, update and modify the data structure, or to interrogate, retrieve and display the data, in the most transparent and convenient way (sect. 4).

The original data structure described in this work, together with its implementation routines, is called MEGA. As described in sect. 2, a decision had to be taken, of course, concerning the basic quantities actually needed to model this specific application. This point is crucial and subject to further discussion when more experience will be gained. In the following we will consider the reader to be familiar with the ER data model ideas.

2. The MEGA data structure

The output of a HEP EG basically consists of a large number of particles of different nature, charge and four-momentum, with

Particle World, Vol. 2, No. 4, p. 108–116, 1991.

their detailed history in terms of parent particles and secondary vertices. In addition, some EG author-dependent piece of information, namely about the ingredients used in the simulation, is usually included.

In order to make the User's life easier when reading different event files (created via different EG runs) with a given analysis program, or in particular with a given tracking program to simulate the response of an experimental apparatus, we propose for any EG output a unique and standard data structure. This is illustrated in fig. 1, which contains the graphical representation, i.e. the so-called "entity-relationship diagram" (ERD) in the ADAMO language [10]. Here follows a detailed description of this ERD.

The MEGA data structure consists of three "subschemata", ***Run***, ***Event*** and ***Static***, which are logically related to each other, as shown by the arrows in fig. 1. The ***Run*** subschema contains the bookkeeping information about the event generation run; ***Event*** provides the event description, in terms of particle kinematics and vertex coordinates; ***Static*** contains the particle codes and properties. Each subschema is structured into different "entity sets" (ESETs), as illustrated in fig. 1.

The **Run** subschema is made of a unique ESET:

- *Run*, whose "attributes" are the EG-run serial number (SerNum), the EG process class (ProcClas), the number of generated events (NumEv), the date of generation (Date), a comment line in free format (Comment), the beam particles identified according to the internal MEGA particle code which will be described later on (Beam1, Beam2), the beam energies in gigaelectronvolts (Energy1, Energy2), the $\sqrt{s}$ centre-of-mass energy in gigaelectronvolts (CmsEnergy), the start and stop addresses of the EG random-number generator (InSeed, OutSeed), and a flag-type attribute (Weighted) indicating whether the events have been assigned a weight during the generation. The ProcClas attribute corresponds to an EG-dependent code, specifying the class of physical processes activated during the generation run (QCD processes, for instance).

The ESETs of the ***Event*** subschema are:

- *Event Header*, containing the header information for each event, with attributes such as the event serial number (SerNum), the EG subprocess code (EvCode), the number associated to the event (RNum), the event weight (Weight), $\sqrt{\hat{s}}$ (SChan) and $\sqrt{\hat{t}}$ (TChan), i.e. the energies at the parton level, in gigaelectronvolts. The EvCode attribute is again specific to each particular EG and indicates the physics subprocess (for instance, gg fusion) according to which the event has been generated. Of course, depending on the subprocess, one of the two partonic energy values will be zero.
- *Kine*, containing the particle kinematics, i.e. its four-momentum (Px, Py, Pz, Energy) in gigaelectronvolts, for all the particles of the event. In addition, a flag-type attribute (Decay) signals the particle decay, if any. The interacting beam particles are also included in *Kine*, and appear as the first two particles of the ESET.
- *Vertex*, containing the coordinates of the particle production vertex (X, Y, Z), in centimetres, for all particles. This vertex can be either primary (interaction vertex) or secondary (decay vertex). The choice of the centimetre unit has been made to be consistent with the widely used GEANT [12] tracking programme (sect. 4).

In the ***Static*** subschema the ESETs are:

- *Particle*, containing a compilation of particle properties. Its self-explanatory attributes are the particle Name, its Mass (in gigaelectronvolts), its Charge (in elementary charge units) and its Lifetime (in seconds). These data are taken from the updated Review of Particle Properties [2]. For new particles (top and Higgs states, for instance), the data are instead derived from worldwide accepted theoretical predictions, namely those currently adopted in the well-known EUROJET [13] package (one of the EGs interfaced via MEGA, as specified in the following). Notice that the actual mass of a generated particle can always be derived from its *Kine* attributes. It should be pointed out that this ESET, which is easily accessible to the User for consultation (sect. 4), will be yearly updated with the most recent experimental and/or theoretical issues, thanks to a newborn collaboration with the Particle Data Group.
- *PartCode*, containing the corresponding particle code compilation for different event generators and tracking programs. In the present release, the ESET is implemented for the following event generators:

EUROJET	[13]	(EuroCod attribute),
LUND	[14]	(LundCod attribute),
HERWIG	[15]	(HerCod attribute),
ISAJET	[16]	(IsaCod attribute);

and for the GEANT [12] tracking program (GeantCod attribute). In addition, the particle code proposed as a standard by the Particle Data Group (which implicitly provides some information about the particle quantum numbers) and already used in the HEPEVT common block, is also included as the StandCod attribute. The total number of particles so far considered, 1349 in the present release, corresponds to all possible particles that are assigned a code in at least one of the four EGs listed above. Notice that, besides normal particles and antiparticles, elementary objects used in the various EGs (such as quarks, diquarks, gluons, clusters, strings, etc.) are also included.
- *Generator*, containing the list of EGs being currently interfaced via MEGA. Its attributes simply are the EG name (Name) and version number (Version).

The "relationships" within the ***Event*** subschema are:

- *ProducedAt*, a relationship between *Kine* and *Vertex*, linking each particle to its production vertex. This is a "many-to-one" (n : 1) relationship (double-headed arrow in fig. 1), since more than one particle is originated from the same vertex, and "partial" on *Kine* side (vertical bar across the arrow in fig. 1), because of the beam particles appearing in *Kine* and having no production vertex. The direction of the arrow, which is obviously irrelevant for a 1 : 1 relationship, is dictated by the fact that only n(≥ 1) : 1 relationships can be defined in the ADAMO logic.

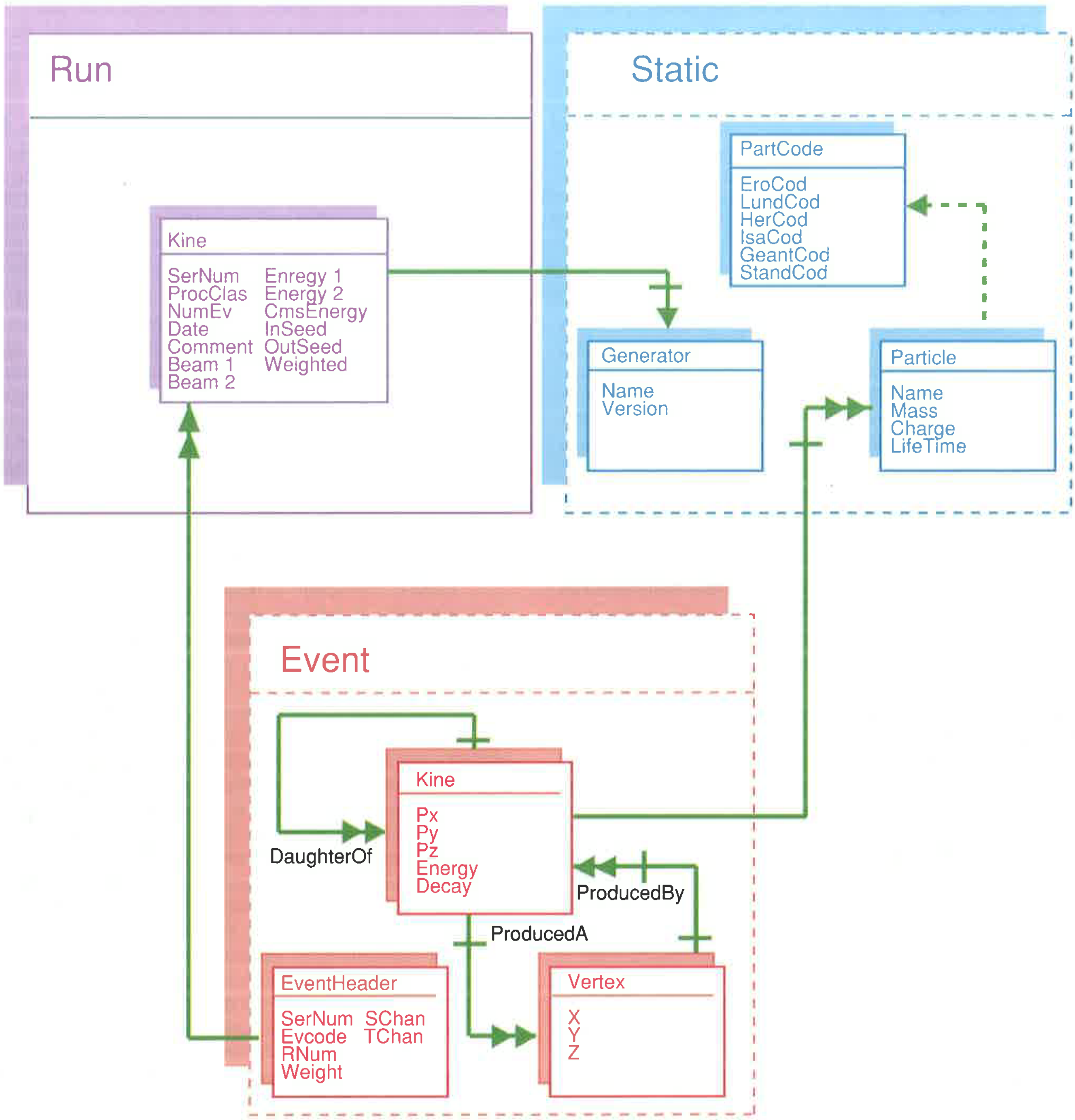

FIGURE 1

Entity-relationship diagram (ERD) relative to the whole MEGA data structure. Subschemata and ESETs are represented as dashed-line and full-line panels respectively, and logical relationships between ESETs as different kinds of arrows (as explained in the text).

• *DaughterOf*, a relationship internal to Kine, linking each decay particle to its parent particle. This is also a partial n : 1 relationship because several particles share the same parent, except for the beam particles.

• *ProducedBy*, a relationship between Vertex and Kine linking each vertex to the particle producing it. This relationship is the reverse of *ProducedAt*. It is n : 1 (since more than one vertex can be associated to a particle, for instance in case of bremsstrahlung emission), and "partial" on both sides (due to the presence of stable particles in Kine, and of the primary interaction vertex in Vertex). Although *ProducedBy* is redundant in the system logic, it has been introduced for practical data presentation purposes, in particular at the program debugging level.

In the ***Static*** subschema, there is just one relationship:

• A 1 : 1 relationship (single-headed arrow in fig. 1) between Particle and PartCode linking each particle to its corresponding codes. This relationship is "implicit" (dashed-line arrow in fig. 1), namely it is achieved in the ADAMO system via the direct use of table row indices (IDs), instead of pointers. These IDs actually define the internal MEGA particle code. Notice that a relationship is explicitly named only in case of ambiguity, otherwise it is assigned by default the name of the ESET which is pointed at.

Additional relationships link ESETs belonging to different subschemata:

• EventHeader (***Event***) is related to Run (***Run***) via a n : 1 relationship, which makes the run bookkeeping information available for each event.

• Kine (***Event***) is in turn related to Particle (***Static***) via a n : 1 relationship, in such a way that all particle properties and codes are available for each generated particle. This relationship is partial on Particle side, since not all existing particles are generated in an event.

• Run (***Run***) is obviously related to Generator (***Static***) in order to know the EG actually used for each particular event-generation run. The relationship is 1 : 1, and partial on Generator side, since not all EGs listed in Generator, but one, will correspond to the current run.

The ER model of MEGA is implemented in the ADAMO system via a simple "data definition language" (DDL). This DDL, relative to Kine and Vertex only, is listed in Appendix A. Partial printouts of Particle and PartCode are given in Appendix B. The ESETs are presented in a suitable table format via the ADAMO handling tools [11].

3. The MEGA package

The MEGA package contains a set of routines (less than 1000 lines of ANSI FORTRAN77 coding, altogether) to read the data from a chosen EG output file and to store them into the appropriate ESETs. The package is handled via the new Code Management system based on ZEBRA (CMZ) [17] (backward compatible with its predecessor, PATCHY [18]). The MEGA data structure has been created using ADAMO. The actual storage of tables is handled by the ZEBRA [19] data-structure management system.

The MEGA application runs either on IBM(VM) or on VAX(VMS) computers. A CRAY(UNICOS) compatible version of the ADAMO system is currently in preparation, thus granting a further increased portability of the MEGA package itself.

Specifically, MEGA interfaces the following event generation programs:

LUND PYTHIA /JETSET(*) (with HEPEVT);
HERWIG (with HEPEVT);
ISAJET;
EUROJET /EURODEC(*) (with HEPEVT).

It should be emphasized that MEGA works for any kind of particle interaction, be it a hadron–hadron collision, an electron–positron annihilation or a deep inelastic lepton–hadron scattering. In particular, it can be used with any other EG of the LUND "family" (LEPTO, AROMA, LUCIFER, TWISTER [14]).

A complete list of MEGA library routines is given in Appendix C.

4. Applications of MEGA

As already pointed out, the application of MEGA in the field of Monte Carlo simulations appears to be particularly attractive. First of all, because a complete transparency relative to the EG internal data representation is achieved. This implies that a unique analysis program is enough, no matter which EG output file is used (sect. 2). Second, because working in the general framework of an ER model, the entire analysis procedure appears to the User as a much simpler job. In this respect, the portability of MEGA (sect. 3) is another serious advantage. The specific application of MEGA for detector simulation purposes is especially relevant now in view of the design of new experiments in future supercolliders, such as LHC, SSC and ELOISATRON (ELN).

The simulation of multi-terra-electronvolt interactions inside a large and complex apparatus is not an easy task. Programs are normally structured into two independent parts: one to simulate the physical event, i.e. the EG, the other to simulate the setup response, i.e. the TP. They both work separately in the sense that in the EG the apparatus where the event takes physically place is not taken into account, while in the TP the generated particles are treated as endproducts of the event. The User is likely to be in charge of the connection between these parts. However, this turns out to be more than a simple interfacing procedure, since most of the time an ultimate generation step has to be completed before the event can actually be tracked in the apparatus.

(*) Fragmentation and decay package.

To illustrate this point, let us consider for instance the decay of long-living particles. Obviously, at the EG level, any decay occurs in the vacuum. In fact these particles, because of their long-flying path, can be affected by the interaction with the various detector components and/or by the presence of a magnetic field, if any. Their behaviour is related to the experimental setup, hence their correct decay simulation must be implemented by the User. Another example comes from short-living particles (Heavy Flavours). Most of the EGs make these particles decay at the primary interaction vertex. Therefore, the correct secondary vertex generation must be supplied by the User, which is essential when vertex detector studies are foreseen.

In order to satisfy the need of an exhaustive and flexible Monte Carlo simulation structure, capable to meet the whole lot of possible User requirements both at the EG and at the TP level, a new software package has been recently implemented: the Full Monte Carlo Chain (FMC) [20]. This connects any EG (sect. 2) with the well-known GEANT [12] tracking program via MEGA. Particular care has been taken to treat the above quoted problems, i.e. short and long-living particle decays, in such a way that the *Kine* and *Vertex* ESETs originally created by the EG are automatically updated in FMC with the correct information provided within GEANT, before the particle tracking is enabled. A sketch of the FMC data flow is presented in fig. 2, where the central role of MEGA clearly shows up.

In addition to MEGA, the detector description used by FMC as input to GEANT is again a standardized ADAMO-based data structure, which can be handled via a dedicated package: the SetUp Descriptor (SUD) [21–24]. This allows the User to define large and complex detectors with a simple table-filling procedure, thus bypassing the usual GEANT procedure which often corresponds, for those who are familiar with this program, to a long coding, difficult to handle, modify or debug. As a further improvement, an interactive version of SUD, where the detector definition can be achieved via a self-explanatory panel-filling procedure, has been implemented and is currently under test [25]. Each panel explicitly displays the GEANT name and meaning of each parameter to be set, so that any detector can be easily configured in the most straightforward way. For details about FMC operation, we refer the reader to the FMC User's Guide [26].

Finally, for an efficient particle data consultation and/or event-by-event analysis, the MEGA data structure can be directly handled using the special ADAMO "tool-kit" (already mentioned in sects 1, 3): TAP (TAble Package) [11], and its interactive version TIP (Table Interaction and Plotting) [27], which has been recently implemented. This kind of application is particularly convenient for program debugging, event display, etc.

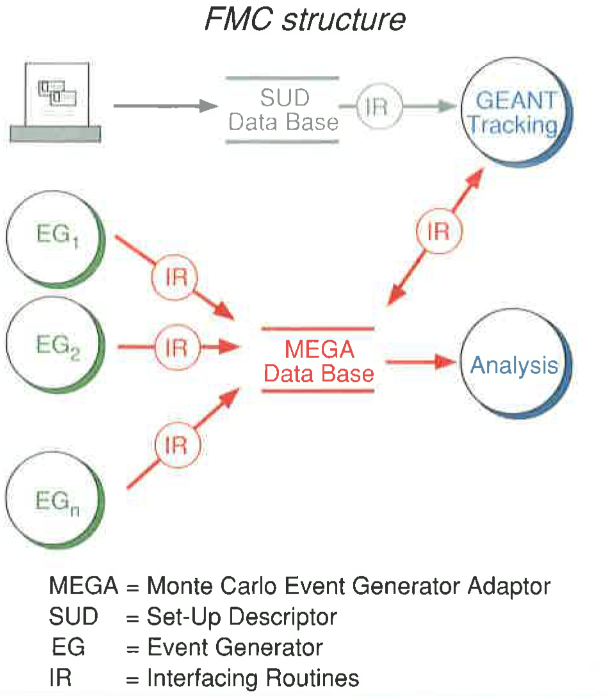

FIGURE 2

A sketch of the FMC data flow via MEGA.

5. Conclusions

In the present work a new software package MEGA to be used in HEP Monte Carlo simulation studies is presented. This package is written with data-modelling techniques, namely the ER model and its FORTRAN implementation (ADAMO). The MEGA package provides a standard organization of different event-generator outputs and finds its major applications both at the detector-simulation level and at the analysis level, with great advantages as far as compactness, modularity, transparency and portability of the program coding are concerned. A first application of MEGA as the input data structure of the GEANT tracking program is already available.

With MEGA we show that a new kind of management of huge and complex data sets is indeed achievable, and suitable to be further successfully used not only in the field of Monte Carlo simulations, but also for real data analysis in future HEP experiments.

Acknowledgements

This work has been supported by the CERN LAA Project (in the framework of the "Supercomputers and Monte Carlo Simulations" component), by the World Laboratory HED (High-Energy Detectors) Project, and by the INFN ELOISATRON Project. We wish to thank P. Palazzi for his very much appreciated contribution, G. D'Alì for his collaboration in the early stage of this work and G. Xèxeo for his help in the DDL optimization.

APPENDIX A

Partial printout of the DDL relative to *Kine* and *Vertex*

```
SUBSCHEMA Event
/*-------------------------------------------------------------------------*/
    : 'Event information'
  AUTHOR     'MSL_Group'
  VERSION    '1.1'
  DATE       '900321'

  DEFINE ESET
/*----------------*/

  Kine   =   (P(4),Decay)
   ALIAS
    PX = P(1);
    PY = P(2);
    PZ = P(3);
     Energy = P(4)
   END ALIAS
       : 'particle four-momentum in GeV/c';

  Vertex   =   (R(3))
       ALIAS
        X = R(1);
        Y = R(2);
        Z = R(3)
       END ALIAS
         : 'vertex position';

  END ESET

  DEFINE RSET
/*----------------*/

       (Kine    [0,1] -> [1,*] Vertex
              BY PRoducedAt)
                     : 'Each four-momentum is produced in a vertex, or is external (beam)';

       (Kine     [0,1] -> [1,*] Kine
              BY DaughterOf)
                     : 'Some particles are generated by other particles';

       (Kine     [1,1] -> [0,*] Particle)
              : 'Each four-momentum is of a given particle type';

       (Vertex   [1,1] -> [0,*] Kine
              BY PRoducedBy)
                     : 'Each vertex is produced by  particle(s) except the beam vertex';

  END RSET

END SUBSCHEMA
```

APPENDIX B

B1. Partial printout of *Particle*

Table : Particle — ADAMO/TAP
Count = 1349

ID	Name	Mass	Char.	LifeTime
1	Down	0.324999	-0.3	–
2	Anti Down	0.324999	0.3	–
3	Up	0.324999	0.7	–
4	Anti Up	0.324999	-0.7	–
5	Strange	0.499999	-0.3	–
6	Anti Strange	0.499999	0.3	–
7	Charm	1.599992	0.7	–
8	Anti Charm	1.599992	-0.7	–
21	Neutrino t.	0.000000	0.0	
22	Anti Neu. t.	0.000000	0.0	–
23	Electron	0.000511	-1.0	0.9999977E+15
24	Positron	0.000511	1.0	0.9999977E+15
25	Muon -	0.105658	-1.0	0.0000022
26	Muon +	0.105658	1.0	0.0000022
27	Tau -	1.784091	-1.0	0.3039994E-12
28	Tau +	1.784091	1.0	0.3039994E-12
29	Gamma	0.000000	0.0	–
30	W +	80.999847	1.0	0.2437778E-24
31	W -	80.999847	-1.0	0.2437778E-24
32	Z 0	92.399841	0.0	0.2612666E-24
54	Pion +	0.139567	1.0	0.2602994E-07
55	Pion -	0.139567	-1.0	0.2602994E-07
56	Pion 0	0.134973	0.0	0.8399983E-16
57	eta	0.548799	0.0	0.6094444E-18
58	Kaon +	0.493645	1.0	0.1237096E-07
59	Kaon -	0.493645	-1.0	0.1237096E-07
60	Kaon 0	0.497670	0.0	–
61	Anti Kaon 0	0.497670	0.0	–
62	Kaon 0 Short	0.497670	0.0	0.8921978E-10
63	Kaon 0 Long	0.497670	0.0	0.5179993E-07
85	phi(1020)	1.019408	0.0	0.2902662E-20
190	Proton	0.938271	1.0	0.1599993E+33
191	AntiProton	0.938218	-1.0	0.1099993E+15
192	Neutron	0.939564	0.0	895.9976807
193	Anti Neutron	0.939483	0.0	895.9976807
194	Lambda	1.115623	0.0	0.2630988E-09
195	Anti Lambda	1.115623	0.0	0.2586988E-09
196	Sigma +	1.189363	1.0	0.7989982E-10
197	Anti Sigma -	1.189363	-1.0	0.7989982E-10
198	Sigma 0	1.192541	0.0	0.7399972E-19

B2. Partial printout of *PartCode*

Table : PartCode — ADAMO/TAP
Count = 1349

ID	StandC	HerCo	IsaCo	EuroC	LundC	Geant
1	1	1	2	2	502	–
2	-1	7	-2	-2	-502	–
3	2	2	1	1	501	–
4	-2	8	-1	-1	-501	–
5	3	3	3	3	503	–
6	-3	9	-3	-3	-503	–
7	4	4	4	4	504	–
8	-4	10	-4	-4	-504	–
21	16	126	15	96	12	4
22	-16	132	-15	-96	-12	4
23	11	121	12	91	7	3
24	-11	127	-12	-91	-7	2
25	13	123	14	93	9	6
26	-13	129	-14	-93	-9	5
27	15	125	16	95	11	34
28	-15	131	-16	-95	-11	33
29	22	59	10	99	1	1
30	24	198	80	198	3	42
31	-24	199	-80	-198	-3	43
32	23	200	90	192	2	44
54	211	38	120	-210	17	8
55	-211	30	-120	210	-17	9
56	111	21	110	110	23	7
57	221	22	220	220	24	17
58	321	46	130	-310	18	11
59	-321	34	-130	310	-18	12
60	311	50	230	-320	19	–
61	-311	42	-230	320	-19	–
62	310	60	20	328	37	16
63	130	61	-20	329	38	10
85	333	56	331	331	35	–
190	2212	73	1120	2110	41	14
191	-2212	91	-1120	-2110	-41	15
192	2112	75	1220	2210	42	13
193	-2112	93	-1220	-2210	-42	25
194	3122	78	2130	3120	57	18
195	-3122	96	-2130	-3120	-57	26
196	3222	86	1130	3110	43	19
197	-3222	104	-1130	3110	-43	29
198	3212	79	1230	3210	44	20

APPENDIX C

List of MEGA library routines

MSMINI	Initialize ADAMO
MSMFIH	Read EG run data and fill ***Run*** subschema
MSMERE	Read EG event
MSMEFL	Initialize ***Event*** subschema
MSMLUN	Fill ***Event*** subschema with PYTHIA 5.5/JETSET 7.3 data
MSMJ73	Fill ***Event*** subschema with PHYTIA 5.5/JETSET 7.3 data, as from HEPEVT common block
MSMHWG	Fill ***Event*** subschema with HERWIG data, as from HEPEVT common block
MSMISA	Fill ***Event*** subschema with ISAJET 6.36 data
MSMEUR	Fill ***Event*** subschema with EUROJET 3.1/EURODEC 2.5 data, as from HEPEVT common block
MSMEFT	Fetch MEGA tables
MSMPRT	Print MEGA tables
MSMCLO	Close ADAMO

References

[1] A. Zichichi, Report on the LAA Project, vol. 1 (1986);
A. Zichichi, Report on the LAA Project, vol. 2 (1987);
A. Zichichi, Report on the LAA Project, vol. 3 (1988);
A. Zichichi, Report on the LAA Project, vol. 4, CERN/LAA 88–1 (1988);
A. Zichichi, Report on the LAA Project, vol. 5, CERN/LAA 88–2 (1988);
A. Zichichi, Rivista del Nuovo Cimento 13/5 (1990) 1;
A. Zichichi, Report on the LAA Project, vol. 6, CERN/LAA 89–1 (1989).

[2] M. Aguilar-Benitez et al., Particle Data Group, Phys. Lett. B204 (1988) 1.

[3] G.R. Lynch and T.G. Trippe, Particle ID numbers, decay tables, and other possible contributions of the particle data group to Monte Carlo Standards, LBL–24287, Proc. of the Workshop on Detector Simulation for the SSC, Argonne, Illinois, USA (1987).

[4] G. Altarelli, R. Kleiss and C. Verzegnassi, Z physics at LEP 1, vol. 3, CERN 89–08 (1989).

[5] A. Putzer, Comput. Phys. Commun., 57 (1989) 156.

[6] S.M. Fisher and P. Palazzi, Comput. Phys. Commun. 57 (1989) 169.

[7] V. Blobel et al., Databases and bookkeeping for HEP Experiments, RL/83–085 (1983).

[8] P.P. Chen, ACM Trans. on Database Systems, 1 (1976) 9.

[9] M.G. Green, The ADAMO Data System: An introduction for particle physicists, RHBNC 89–01 and CERN/DD/US/131 (1989).

[10] R. Brazioli et al., ALEPH Collaboration, The ADAMO programmer's manual, ADAMO Notes (1988).

[11] L. Basadonna et al., ADAMO TAP, The TAble Package programmer's reference manual, ADAMO Note 6 (1987).

[12] R. Brun et al., GEANT3 User's Guide, CERN/DD/EE 84–1 (update Sept. 1987).

[13] F. Anselmo and B. van Eijk, EUROJET vers. 3.1 (in prep.);
A. Ali and B. van Eijk, EURODEC, CERN Pool Program W5048 (1990).

[14] The Lund Monte Carlo Programs, CERN Pool Program W5035/W5046/W5047/W5048 long write-up (1987) (and references therein);
H.U. Bengtsson and T. Sjöstrand, Comput. Phys. Commun. 46 (1987) 43.

[15] G. Marchesini and B.R. Webber, HERWIG, CERN Pool Program W5037 (1990).

[16] F.E. Paige and S.D. Protopopescu, ISAJET, CERN Pool Program W5036 (1990).

[17] M. Brun et al., Comput. Phys. Commun. 57 (1989) 235.

[18] H.J. Klein and J. Zoll, PATCHY Reference Manual, CERN Program Library (1988).

[19] R. Brun and J. Zoll, ZEBRA User's Guide, CERN Program Library Q100 (1987).

[20] C. La Commare and M. Marino, FMC: a full Monte Carlo chain for HEP experiments CERN/LAA–MSL 91–06 (to be submitted to Comput. Phys. Communications).

[21] A. Bassi et al., GEANT–ZEUS–FMUON data structure with ADAMO, ADAMO Application Note (1987);
A. Bassi et al., ZEUS Internal Note 88–7 (1988);
A. Bassi et al., A suggestion for the ZEUS geometry definition model and for the FMUON data model, ZEUS Internal Note 88–97 (1988).

[22] R. Nania, Monte Carlo simulation for the LAA Detector, proc. of the 9th Workshop of the INFN ELOISATRON Project on Perspectives for New Detectors in Future Supercolliders, Erice, Italy (1989).

[23] G. Anzivino et al., A general definition of a HEP apparatus based on the ADAMO package in the framework of the GEANT Monte Carlo program, proc. of the International Workshop on Software Engineering, Artificial Intelligence and Expert Systems for High-Energy and Nuclear Physics, Lyon, France (1990).

[24] A. Bassi et al., Detector simulation for complex multi-TeV experiments, CERN/ECP 91–18, and proc. of the XIIth Workshop in New Technologies for Super colliders, Erice, Italy (1990), Plenum Pub. Co., London (to appear).

[25] G. La Commare and M. Marino, SUD–IS User's Manual, version 1.0, CERN/LAA–MSL 91–12.

[26] J. Alberty et al., FMC User's Guide, version 1.2, CERN/LAA-MSL 90–01 (1990).

[27] J.M. Fisher and P. Palazzi, The ADAMO data system programmer's manual, version 3.1 (1991).

Addresses:

F. Anselmo, O. Di Rosa, G. La Commare and A. Zichichi
CERN/LAA
CH–1211 Geneva (Switzerland)

E. Barberio and M. Marino
World Laboratory/HED Project
6, Place de la Riponne
CH–1005 Lausanne (Switzerland)

G. Bruni, P. Bruni, L. Cifarelli, F. Cindolo, G. Iacobucci and R. Nania
INFN
I–40126 Bologna (Italy)

L. Cifarelli
Physics Department
University of Naples
I–80125 Naples (Italy)

Received on April 1991.