Chapter C

Methods and tools

1 Heritage projects, preservation, and re-usability concerns

Contribution* by: S. Banerjee, M. Chrzaszcz, Z. Was, J. Zaremba
Corresponding author: Z. Was [z.was@cern.ch]

The FCC is a long-term project, novel in many respects, and new calculations, including simulation programs, will be appearing in the forthcoming years. However, many of the approaches developed for previous experiments, in particular LEP, will be useful, either directly as a tools or as a means to prepare substantial benchmarks. In addition, programs that will be prepared for Belle II, especially in the domain of τ, B, and D resonance physics, will continue to be valuable tools. Such programs and projects will undoubtedly evolve in the meantime, but one can expect that ready-to-use versions will be available when the need arises. Then only interfaces will need to be archived solely for the FCC. In some cases, the whole projects will require long-term preservation. Before we will explain some attempts on preservation of some example projects, such as τ decays, radiative corrections in decays, or electroweak corrections, let us mention possible general approaches.

There are many helpful tools for managing software projects, in both development and preservation. However, preservation-development tools become obsolete and code history, necessary for future extensions and validation, may become lost; therefore, it is important to ensure proper migration from one preservation-development tool to the next. In addition, very stable solutions belong to repositories beyond an author's responsibility, specifically targeting long-term preservation.

The CPC International Program Library [1] serves such a purpose; CERN web pages like that used for TAUOLA [2] or PHOTOS [3] may also offer the necessary facility.

Issues arise if parts of the code are prepared using automated code development tools. If those tools (i.e., other programs) are not published, the programs prepared with their help are of limited help for future applications, especially if extensions are needed.

The interfaces between segments of the code can be of a different type. The reassurance offered by solutions based on some tools is indisputable, but can be overshadowed if such a tool evolves over an unsuitable time. We have experienced minor, but at inconvenient moments for our project evolution, difficulties as a result of ROOT library [4] upgrades to new versions. Manual intervention on our part and changes to the work routine were necessary.† Because many software development projects of phenomenology represent a sizeable fraction of the total effort

*This contribution should be cited as:
S. Banerjee, M. Chrzaszcz, Z. Was, J. Zaremba, Heritage projects, preservation, and re-usability concerns, DOI: 10.23731/CYRM-2020-003.135, in: Theory for the FCC-ee, Eds. A. Blondel, J. Gluza, S. Jadach, P. Janot and T. Riemann, CERN Yellow Reports: Monographs, CERN-2020-003, DOI: 10.23731/CYRM-2020-003, p. 135.

†We had to modify the code for our projects, owing to changes in the ROOT library, first in July 2002 and again for the changes introduced with ROOT release 6.

and the people involved may often not be immediately available, this may represent a major inconvenience.

1.1 Common tools for all FCC design studies

As already mentioned, it is of crucial importance to have a common software platform with all the repositories. In the FCC, this effort has begun with the creation of a twiki page [5], where different MC generators are available. This collection should be extended with documentation of programs, links to available original `git` repositories, etc. Everybody is welcome to link or put there related codes or results to be used in future software.

The twiki page [5] currently includes three sections.

1. *FccComputing*, in which the installation procedure of the FCC-ee software is described.
2. *FccGenerators*, containing different MC generators. Currently, Tauola, Higgsline, KKMC, and Bhabha generators are presented.
3. *FccSoftware*, containing various examples of simulations run in the FCC framework.

Next, we will briefly describe currently available generators and discuss their preservation.

1.1.1 Tauola

The τ decay phenomenology relies to a large degree on experimental data. This is because of the complexity of experimental analyses and the difficulty in phenomenology modelling decays where intermediate resonances used in hadronic currents are broad and perturbative QCD description is only partly suitable. Background analysis for multidimensional distributions is a problem. Collaborations are hesitant to enable outside use of their matrix element parametrizations, because they may be unsuitable for other, externally studied, distributions. Nevertheless, if they become available, it is worthwhile to store them in publicly available repositories. In Ref. [6], parametrizations developed for ALEPH and CLEO were archived, together with the original parametrization, useful for technical testing of the Tauola algorithm. In Ref. [7], thanks to discussions with the BaBar community, an extension of Tauola with multichannel capacity that is easy to manipulate by users was prepared. The resulting default parametrization equivalent to that work was archived in Ref. [8]. In that reference, a framework for work with C++ currents and for Belle II was prepared. Hopefully, this may provide a means to feed back code at the FCC. A smooth transition period for evolution from partly Fortran to fully C++ code is envisaged in this solution.

1.1.2 Photos and Tauola Universal interface

The code for these projects is currently in C++. Preservation of the up-to-date variants is assured, thanks to CERN special accounts and web pages [2, 3]. Some versions are archived in the CPC [9,10]. The main issue for the project is the fast evolution of event format HepMC [11], and especially how other projects use that format to write down generated events. In addition, long-term preservation efforts may be impaired because of the evolution of configuration and make-file arrangements.

1.1.3 EvtGen

The decays of heavy flavoured hadrons provide huge constrains on Beyond Standard Model physics [12]. The FCC-ee is due to run on the Z pole, and will also be a heavy flavour factory. The decays of such mesons and hadrons are modelled with the `EvtGen` package [13]. The package consists of various models, which are constantly being updated with the theory predictions, such as form factors and amplitude calculations. Currently, the main developers of `EvtGen` are involved in LHCb collaboration; however, the package is made publicly available [14] via the git repository. It was recently extended to describe the decays of spin 1/2 particles. The project is written in C++ and interfaces with the HepMC format [11]. It is also possible to interface it with the `Tauola`, `Pythia`, and `Photos` packages.

1.1.4 Electroweak corrections

The `KKMC` code is published and archived in Ref. [15]. Its electroweak correction library, in use until today, is also published and archived: `Dizet` version `6.21` [16, 17]. At present, only `Dizet` version 6.42 [18, 19] is available for the `KKMC` electroweak sector upgrade. This version of `Dizet` is missing updates, owing to the photonic vacuum polarisation, e.g., as provided in Refs. [20, 21]. We could implement the updates ourselves because `Dizet` version 6.42 is well-documented. However, some versions of `Dizet` that still exist may become unavailable at a later date. In fact, it was difficult for us to obtain access and we decided to revert to version 6.42. This indicates the necessity for code maintenance, even if authors may at some time become unavailable.

In any case, `Dizet` version 6.42 [18, 19] with updates from Refs. [20, 21] is prepared as a facility for the electroweak tables used in `KKMC` [15].

Using tables prepared by one program in another program is not only the method to enhance the speed of the calculation. Interpolation of values enable technical regularisation of the functions. Technical instabilities at the phase space edges can be regulated.

The tables can be used by other programs that understand the format. In this way, for example, the `TauSpinner` package [10, 22] can be used for graphical presentation for different variants of `Dizet` and of its initialization as a natural continuation of work [23] for the LHC or similar activities for the FCC.

This limit is a substantial burden for interfaces. Preservation of projects is only partly assured by the CPC publications. The most up-to-date versions are available at user webpages, which sometimes are not available or may become unavailable.

References

[1] http://www.cpc.cs.qub.ac.uk/, last accessed 22 January 2020.

[2] http://tauolapp.web.cern.ch/tauolapp/, last accessed 22 January 2020.

[3] http://photospp.web.cern.ch/photospp/, last accessed 22 January 2020.

[4] R. Brun and F. Rademakers, *Nucl. Instrum. Meth.* **A389** (1997) 81. doi:10.1016/S0168-9002(97)00048-X

[5] https://twiki.cern.ch/twiki/bin/view/FCC/CommonTools, last accessed 22 January 2020.

[6] P. Golonka *et al.*, *Comput. Phys. Commun.* **174** (2006) 818. arXiv:hep-ph/0312240, doi:10.1016/j.cpc.2005.12.018

[7] Z. Was and P. Golonka, *Nucl. Phys. Proc. Suppl.* **144** (2005) 88. arXiv:hep-ph/0411377, doi:10.1016/j.nuclphysbps.2005.02.012

[8] M. Chrzaszcz *et al.*, *Comput. Phys. Commun.* **232** (2018) 220. arXiv:1609.04617, doi:10.1016/j.cpc.2018.05.017

[9] N. Davidson *et al.*, *Comput. Phys. Commun.* **199** (2016) 86. arXiv:1011.0937, doi:10.1016/j.cpc.2015.09.013

[10] N. Davidson *et al.*, *Comput. Phys. Commun.* **183** (2012) 821. arXiv:1002.0543, doi:10.1016/j.cpc.2011.12.009

[11] M. Dobbs and J.B. Hansen, *Comput. Phys. Commun.* **134** (2001) 41. doi:10.1016/S0010-4655(00)00189-2

[12] J. Aebischer *et al.*, *Eur. Phys. J.* **C80** (2020) 252. arXiv:1903.10434, doi:10.1140/epjc/s10052-020-7817-x

[13] https://evtgen.hepforge.org/, last accessed 22 January 2020.

[14] https://phab.hepforge.org/source/evtgen, last accessed 22 January 2020.

[15] S. Jadach *et al.*, *Comput. Phys. Commun.* **130** (2000) 260. arXiv:hep-ph/9912214, doi:10.1016/S0010-4655(00)00048-5

[16] D.Y. Bardin *et al.*, *Comput. Phys. Commun.* **59** (1990) 303. doi:10.1016/0010-4655(90)90179-5

[17] D.Y. Bardin *et al.*, *Comput. Phys. Commun.* **133** (2001) 229. arXiv:hep-ph/9908433, doi:10.1016/S0010-4655(00)00152-1

[18] A. Andonov *et al.*, *Comput. Phys. Commun.* **181** (2010) 305. arXiv:0812.4207, doi:10.1016/j.cpc.2009.10.004

[19] A. Akhundov *et al.*, *Phys. Part. Nucl.* **45** (2014) 529. arXiv:1302.1395, doi:10.1134/S1063779614030022

[20] H. Burkhardt and B. Pietrzyk, *Phys. Rev.* **D72** (2005) 057501. arXiv:hep-ph/0506323, doi:10.1103/PhysRevD.72.057501

[21] F. Jegerlehner, *EPJ Web Conf.* **218** (2019), 01003. arXiv:1711.06089, doi:10.1051/epjconf/201921801003

[22] Z. Czyczula *et al.*, *Eur. Phys. J.* **C72** (2012) 1988. arXiv:1201.0117, doi:10.1140/epjc/s10052-012-1988-z

[23] E. Richter-Was and Z. Was, *Eur. Phys. J.* **C79** (2019) 480. arXiv:1808.08616, doi:10.1140/epjc/s10052-019-6987-x