CMD3FWK SOFTWARE DEVELOPMENT FRAMEWORK

F. Ignatov, S. Pirogov*, A. Sibidanov, A. Zaytsev#,
Budker Institute of Nuclear Physics†, Novosibirsk, Russia
I. Fedko, A. Ognev, Novosibirsk State University‡, Novosibirsk, Russia

Abstract

This contribution contains the general design overview and a status of implementation of the core tools for the CMD-3 detector simulation and event reconstruction. Software design standards of the project are object oriented programming techniques, C++ as a main language, modular approach and XML and XML Schema usage for configuring software components. The dedicated software development framework (Cmd3Fwk) was implemented in order to be the basic integration solution for building offline reconstruction code, simulation tools and the 3rd level trigger for the CMD-3 detector. The framework and a set of modules are currently supported on Linux (Fedora Core 4.x and Scientific Linux 4.x) with GCC compilers up to v4.0 for both x86 and x86_64 architectures.

The offline environment built is highly integrated with the ROOT framework (the persistency management for the reconstructed data is implemented via ROOT), Geant4 toolkit, CLHEP and GSL libraries. All the tools produced are designed and optimized for the batch processing mode and GRID environment compatible.

VEPP-2000 COLLIDER AND CMD-3 DETECTOR

The CMD-3 is the general purpose cryogenic magnetic detector [1, 2] for the VEPP-2000 electron-positron collider [3, 4], which is being commissioned at the Budker Institute of Nuclear Physics (BINP, Novosibirsk, Russia). The main aspects of the physical program of the experiment are the study of known and the search for a new vector mesons, study of the $p\bar{p}$ and $n\bar{n}$ production cross sections in the vicinity of the threshold and search for exotic hadrons in the region of center-of-mass energy below 2 GeV. The VEPP-2000 collider also is going to perform the first test of round beam technique [5, 6].

The essential upgrade of the CMD-2 detector (designed for the VEPP-2M collider at BINP) production farm and distributed data storage management software is being performed in order to satisfy the new detector needs.

*E-mail: S.A.Pirogov@inp.nsk.su
#E-mail: A.S.Zaytsev@inp.nsk.su
†Main Web Site: http://www.inp.nsk.su
‡Main Web Site: http://www.nsu.ru

CMD-3 SOFTWARE DEVELOPMENT FRAMEWORK

General Overview

The CMD-3 Software Development and Data Processing Framework (officially named as *CMD-3 SD/DP Fwk* or simply *Cmd3Fwk*) design is based on the following assumptions about typical HEP data treatment procedure:

- The data analysis procedure is well represented by a directed acyclic graph with the modules and data instances at the nodes, therefore the reverse call method of building self-organizing modules chain can be used.
- The input data is divided into so called ”runs” consisting of so called ”events” with the similar structure withing the certain run, thus the cycle over the events and runs for the data set being analyzed can be organized by the framework tools themselves, not by some code provided by the user.
- The data instances are produced by the modules and only the creator module of the certain data instance is intended to modify it, so all the intermediate stages of the data processing are preserved during the certain event reconstruction.

Hierarchy Data Processing MetaFramework

Recently, the core components of the Cmd3Fwk was moved to the separate detector independent software package named as *Hierarchy Data Processing MetaFramework* (or simply *h-dp-fwk*) with the aim to enable its usage outside the scope of the CMD-3 project. The key features of the MetaFramework are modularity, dynamic data processing chain generation according to the XML configuration of modules and the on-demand data request mechanisms. It also provides the command-line and the graphical user interfaces for building XML configurations and running the data processing jobs. The h-dp-fwk MetaFramework is a powerful tool which can be used for development of the specialized adaptive data processing tools for various applications, for instance, for building small and medium scale HEP experiment specific data processing frameworks.

The key properties of the current h-dp-fwk implementation are listed below:

- The coding style of the framework differs significantly from the official CMD-3 coding style.
- No dependencies on CMD-3 specific software components.

- It requires the libxml2, BOOST and log4cxx libraries to be installed.
- All the external software interfaces are wrapped by the dedicated interfaces of the framework.
- The STL containers are extensively exploited within the core via wrapper interfaces. The associative container wrappers are aware of the regular expressions.
- Modules integration and the message exchange solution is service-oriented with the direct module-to-module dependencies allowed.

Please refer to the Fig. 1 for the detailed schema of the h-dp-fwk architecture.

Current Status of Implementation

The implementation of the Cmd3Fwk has been finished and its production version now is **1.3.5** which is stored in the private CMD-3 collaboration CVS repository. The first production versions of the h-dp-fwk MetaFramework and the Cmd3Fwk **v2.x** based on it are expected in 2006Q2–2006Q3.

The h-dp-fwk will be distributed as a set of RPM packages in order to simplify the initial deployment and support of the CMD-3 offline computing environment. Both Doxygen documentation and the design document are available for the framework components. The user manual and the online TWiki-based manual are being produced.

CONCLUSION

The dedicated software development framework was implemented in order to be the basic software integration solution for building offline reconstruction code, simulation tools and the 3rd level trigger for the CMD-3 detector. Although the main area of application of the framework is supposed to be HEP data processing, it also can be used in any activities related to sequential data processing.

ACKNOWLEDGEMENTS

The authors are grateful to all the members of CMD-3 online and offline software development groups for fruitful discussions during the design stage and also for useful testing and debugging feedback.

REFERENCES

[1] D.N. Grigoriev, CMD-2 Detector Upgrade. **hep-ex/0106009**

[2] CMD-3 Collaboration Web Site: http://cmd.inp.nsk.su

[3] I.A. Koop, VEPP-2000 Project. **physics/0106013**

[4] VEPP-2000 Collaboration Web Site: http://vepp2k.inp.nsk.su

[5] L.M. Barkov et al., Proc. of the IEEE Particle Accelerator Conference, San Francisco (1991), p.183

[6] V.V. Danilov et al., Proc. of the Asian Particle Accelerator Conference, Tsukuba (1998), p.257

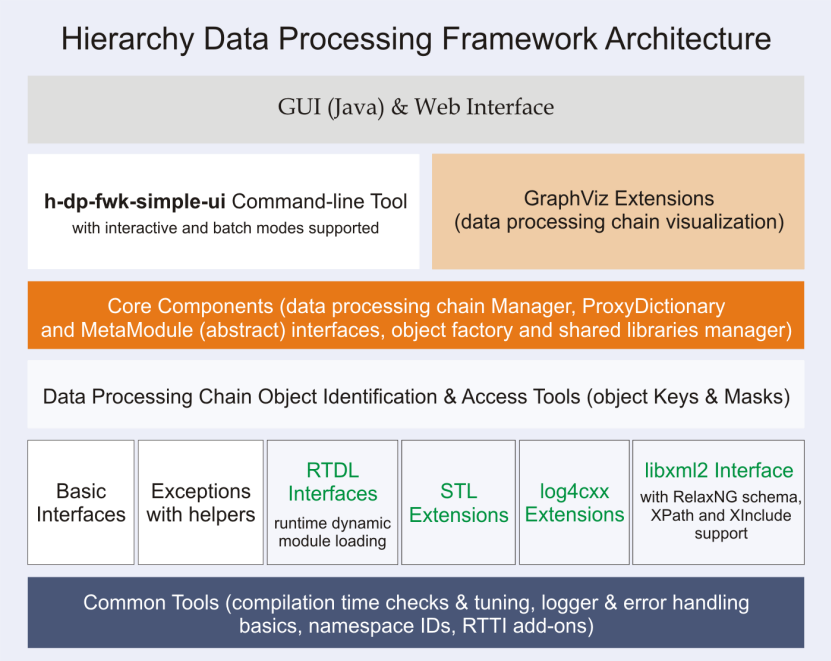

Figure 1: Hierarchy Data Processing MetaFramework (h-dp-fwk) architecture overview.