NA61 Software Upgrade

The NA61/SHINE Collaboration

January 13, 2011

Abstract

This document presents a proposal for a software upgrade of the NA61/SHINE experiment at the CERN SPS. First, the current NA61 software is briefly presented and needs for the software upgrade are discussed. Second, main properties of the proposed new software are outlined. The new software will be based on the C++ language, standard C++ (STL) data containers, flexible I/O (ROOT), and modular structure backward-compatible with the old framework. A similar approach was successfully used in the off-line software of the Pierre Auger Observatory. The new software shall be used within the virtual machine system currently developed by CERN for NA61. This will guarantee its long-term stability and the reproducibility of physics results over the long duration of the NA61 data taking and analysis.

1 Introduction

NA61/SHINE (SHINE = SPS Heavy Ion and Neutrino Experiment) [1] is an experiment at the CERN SPS using the upgraded NA49 hadron spectrometer. Among its physics goals are precise hadron production measurements for improving calculations of the neutrino beam flux in the T2K neutrino oscillation experiment as well as for more reliable simulations of cosmic-ray air showers. Moreover, p+p, p+Pb and nucleus–nucleus collisions will be studied extensively to investigate properties of the onset of deconfinement and search for the critical point of strongly interacting matter.

NA61 started data taking in 2007 and will continue to collect data until at least 2014. Active work on data analysis, reconstruction and simulation is therefore expected to last until the end of the current decade. Due to the shortcomings of the current analysis software (see Section 2), we propose to modernize the NA61 software framework in order to achieve and maintain full flexibility, functionality and portability throughout this period.

2 Current NA61 Software Framework

The current software framework of NA61 has been inherited from its predecessor, NA49. The NA61 software group invested considerable effort to reuse the existing code and to develop missing parts. The NA61 software was set up at CERN using the lxplus/lxbatch cluster with the Scientific Linux (SLC) 4 OS. The software has recently been migrated to version 5 of SLC.

The core of the present framework was developed in the early 1990s and the difficulties in maintaining and extending this software clearly impede the NA61 data processing and analysis. The major shortcomings of the current setup are sketched in the following.

- **Several concurrent data formats:** The internal data format used within the simulation and reconstruction chain are DSPACK [5] structures. Although foreseen as a generic object-oriented way of storing data, this format is not used in the high-energy physics community any more, and therefore it is difficult to get support for it (*e.g.* when dealing with problems of portability due to platform dependence of the data structures). The DSPACK-based DST[1] output of the reconstruction chain is then converted to ROOT [2] based miniDSTs[2] for analysis, with a much smaller amount of detail. Due to these concurrent data formats, most of the analysis software has to be available both for ROOT and for DSPACK, as for deeper data debugging one always needs to go back to the DSPACK level. Apart from the obvious overhead of data and code duplication, the existence of two data formats hampers the collaboration of analysis people and students that either work with DSPACK or ROOT.

- **Different programing languages:** The simulation and reconstruction modules of the current framework were mostly written in FORTRAN and C. Upgrading and maintaining such a multi-language framework has become a bottleneck for the NA61 software community. The FORTRAN code is particularly difficult to maintain due to this language having got phased out by more modern languages; furthermore, a special flavor of FORTRAN is used in NA61, which uses non-standard language features not compatible with F77 or F90 and therefore requires a special compiler. Moreover, since both languages are considered old-fashioned nowadays, it is difficult to motivate students or younger researchers to write new code or even understand existing one. Modern languages (such as C++) offer a better compromise for performance computing within an efficient and comfortable programming framework, for most numerical and data handling problems.

- **Obsolete parts in data model:** The data model of the old framework is highly rigid. On the one hand, some of the sub-detectors of the NA49 experiment which are not used in NA61 have to remain supported by the data structures, as removing them would require a major overhaul of the whole reconstruction chain. On the other hand, introduction of new detectors was not foreseen in the initial design of the data structures and therefore leads to numerous kludges and inconsistencies in the code. This has already been observed in the case of the so-called GAP-TPC (added to the experimental setup in the late stage of NA49) and is very likely to happen again and again as new detectors are added to NA61.

- **Clients:** The reconstruction and simulation chain of NA49 was divided into so-called clients, which are stand-alone executables that read and write data following the client-server paradigm. While such a modular approach ought to be preserved the actual implementation needs an overhaul, since the steering of the clients within a shell script (as it is implemented now) and using command-line arguments as well as environmental shell variables is too error-prone. Moreover, it is difficult to follow which client modifies what piece of the data structures exactly.

- **GEANT3:** The NA61 detector simulation is currently based on GEANT3 [4]. Apart from the aforementioned difficulty to motivate students or younger analysis people to

[1] DST (Data Summary Tape) is referred to as well as ESD (Event Summary Data) in the newer jargon.
[2] MiniDST is referred to as well as AOD (Analysis Object Data) in the newer jargon.

implement new detectors in a not widely used language (FORTRAN), GEANT3 does not meet today's physics requirements any more. For instance, the quality of the hadronic generators implemented in GEANT3 will not be sufficient to model the hadronic showers in the new NA61 sub-detector PSD.

- **Testing:** There is no testing infrastructure integrated with the current framework. As a result, any changes (caused not only by modifications to the framework itself but also by upgrades of libraries, compilers and other components of the underlying operating system) observed in its output have to be debugged by hand, consuming time and resources. What is more, such an approach results in highly non-uniform testing coverage of the framework, leading to non-fatal but possibly significant bugs persisting in the code.

- **Documentation and support:** Documentation of the current framework is scant and largely out of date. Consequently, the working knowledge of the framework has to be re-acquired through analysis of the source code.

3 Proposal for an Upgrade of the NA61 Software Framework

Given the shortcomings of the existing software we propose to write a new software framework for simulation, reconstruction and data analysis. One of the main points to be achieved is the unification of data structures and programming languages. The modularity of the current framework should be preserved and a fast transition to first physics production is mandatory. The main improvements that are envisaged are as follows.

- **C++ language:** We propose to use C++ as the only programming language to be used in the new framework. We choose this language because of its ubiquitous use within the HEP community and its multi-paradigm programming approach. Since we intend to avoid a multi-language framework as much as possible, we find this a good compromise.

- **Unified data structures:** Events will be structured using STL [3] data containers so that well-known and simple objects are used, and the event data shall be streamed using ROOT to ensure flexible and platform-independent I/O. Different levels of details, *i.e.* full reconstruction results and MiniDSTs, will be realized by simply omitting parts of the (more detailed) data while at the same time preserving the overall data structure.

- **Modularity:** For the general layout of the framework, we will follow the approach of [7]. The framework will comprise three principal parts: a collection of processing *modules* which can be assembled and sequenced through instructions provided in an XML file, an *event* data model through which modules can relay data to one another and which accumulates all simulation and reconstruction information, and a *detector description* which provides a gateway to data describing the configuration and performance of the experiment. This approach of sequencing of the processing modules which communicate through an event-store interface decouples the data from the algorithms, thus satisfying the requirements of physicists whose primary objective is collaborative development and refinement of algorithms.

- **Geant4:** The current detector simulation set-up will be migrated to Geant4 [6]. Using existing tools to convert GEANT3 detector descriptions to Geant4 we will be able to reuse the work and validation of the old simulation by the NA49 collaboration.

- **Client wrappers:** An important feature of this proposal is to assure a quick migration from the old to the new framework. This will be achieved by providing wrapper classes which will allow to run the old clients as modules in the new framework. These wrappers will provide the necessary conversion from the new to the old data structures and vice versa. Being able to run the old physics code in the new framework will not only allow starting data production as soon as the basic functionality of the framework is ready, but also help ensure the validity of the new software by being able to compare results produced by the two frameworks using identical algorithms. Once the compatibility of the frameworks will have been verified, the clients will be replaced one by one, depending on the need for new algorithms or refinement of the old ones.

- **Automated testing:** the new framework will feature an integrated fine-grained testing system offering unit, coverage and stress tests. It will allow both monitoring the performance of the framework and pinpointing the source of the discrepancies/errors, with as little human intervention as possible. Each new module will be required to provide appropriate self-tests.

- **Documentation:** A dual procedure will be implemented to facilitate maintaining thorough and up-to-date documentation of the framework. First of all, as much documentation as possible will be automatically generated from the source code and comments within it. On the other hand, to account for cases where automated generation is not feasible, the framework's release procedure will explicitly require documentation of newly included or modified components to be present before they can be incorporated into the mainline source tree.

Since mid-2010, work has been ongoing on virtualization of the NA61 software and running the data-reconstruction framework within virtual machines [8]. It is therefore imperative to outline the relation of the proposed software upgrade to the ongoing virtualization project. First, it is planned to install the new NA61 software under the CernVM system and use it for the simulation and reconstruction of data. This will from the beginning allow us to avoid effort related to the fast changes of the software environment (operating systems, compilers, . . .) and assure its long-term stability. This will be of particular importance after the development phase of the new software, when the number of software experts within NA61 is expected to be significantly reduced. Furthermore, the use of the new software under the CernVM will guarantee long-term reproducibility of the results. This is of a great importance for the NA61 physics program, which has to rely on analysis of data collected over many years. Therefore, both projects are complementary.

4 Manpower and Resources

The proposed software upgrade requires a core group (2-3 seniors, 2-3 students) working together at CERN for an extended period of about one year. The list of participants of the core group at CERN is given in Table 1.[3] The work is foreseen to start in March 2011 and the duration of stay at CERN of participants will vary from two months to one year.

Name	Status	Institute	Financial support
András LÁSZLÓ	senior	CERN, PH-SME (research fellow)	CERN Fellow
Michael UNGER	senior	Karlsruhe Institute of Technology	2 months NA61 CF
Darko VEBERIČ	senior	University of Nova Gorica	2 months NA61 CF
Antoni MARCINEK	Ph.D. student	Jagellonian University, Krakow	1 month EU Ph.D. Fellowship
Roland SIPOS	M.Sc. student	Eötvös University, Budapest	12 months not solved
Oskar WYSZYŃSKI	M.Sc. student	Jagiellonian University, Krakow	12 months not solved

Table 1: Participants of the NA61 software upgrade core group at CERN. Sources of their financial support at CERN are also indicated.

5 Request for CERN Support

In order to achieve an efficient, timely start and continuation of the work on the software upgrade, support from CERN is necessary. Namely, support for a stay at CERN of two technical students (12 months each) would solve the funding problems referred to in Table 1.

References

[1] N. Antoniou *et al.* [NA61/SHINE Collaboration], CERN-SPSC-2007-004, (2007); CERN-SPSC-2007-019, (2007); N. Abgrall *et al.* [NA61/SHINE Collaboration], CERN-SPSC-2008-018, (2008).

[2] ROOT: `http://root.cern.ch`

[3] STL (Standard Template Library for C++): `http://sgi.com/tech/stl`

[4] R. Brun *et al.*, "GEANT3 Users Guide", CERN Report DD/EE/84-1 (1987).

[5] R. Zybert, P. Buncic, "DSPACK: Object manager for high energy physics" Proc. CHEP95 (1995), 345.

[6] S. Agostinelli *et al.*, "GEANT4: A Simulation toolkit", Nucl. Instrum. Meth. **A506** (2003) 250.

[7] S. Argiro *et al.*, "The Offline Software Framework of the Pierre Auger Observatory", Nucl. Instrum. Meth. **A580** (2007) 1485.

[8] Software Virtualization:
`http://indico.cern.ch/getFile.py/access?contribId=43&resId=1&materialId=slides&confId=109577`

[3] For completeness of the list of CERN resources used for NA61 related software activities, we mention that Mihajlo Mudrinic works as a project associate on the Software Virtualization project. As this activity is decoupled from the proposed software upgrade of NA61, we do not list him in Table 1.