THE OFFLINE SOFTWARE FRAMEWORK OF THE NA61/SHINE EXPERIMENT
Roland Sipos, for the NA61 Collaboration
E-mail: Roland.Sipos@cern.ch
Co-authors: Andras Laszlo, Antoni Jerzy Marcinek, Tom Paul, Marek Szuba, Michael Unger, Darko Veberic, Oskar Wyszynski

Introduction
NA61/SHINE (SPS Heavy Ion and Neutrino Experiment) is an experiment at the CERN SPS using the upgraded NA49 hadron detector. Moreover, p+p and nucleus-nucleus collisions will be studied extensively to allow for a study of properties of the onset of deconfinement and search for the critical point of strongly interacting matter.

Currently NA61/SHINE uses the old NA49 software framework for offline related tasks. The core of this legacy framework was developed in the early 1990s and the collaboration made huge efforts to maintain the rigid software environment, without great success. In this contribution we will introduce the new software framework, called shine, a tool for data reconstruction, simulation and analysis.

Legacy software
DPSACK is an implementation of a client-server software architecture with extended capabilities, such as combinations of early object oriented principles. There are multiple choices to structure and store a given data set and provide as a resource for requesters. The ideology was that kind of server application should handle all data requests and store the description of physical event. Basically every standalone program that attempts to connect to a DPSACK server instance and manipulates stored data, are called clients. On can create standalone programs that use this memory manager to solve sub-tasks of an application. The application itself will be the chain of client calls and DPSACK instructions in a script. In the legacy software several related complex tasks were represented with the chain ideology. There are multiple services provided by CERN, on which the NA61/SHINE depends. A complicated environmental setup is required in order to start processing, such as setting up the directory of binaries or paths to dependencies. The environment of the legacy heavily relies on the /fixed paths provided by CERN IT infrastructure.

Software upgrade proposal
The experiment’s data taking period started in 2007 and will continue to collect data until 2014. An agile and portable software environment is rather a need than an option to aid the continuous productive work for the Collaboration. A decision on a software upgrade strategy was taken, taking into account the necessary changes for the experiment.

Pattern and skeleton
The Auger Offline Framework was used as the skeleton for Shine. The design contains distinguished key features, such as extensibility, and there are three principal pillars describing the planned framework.

Configuration
Central configuration stands as the interface to every configuration related task. The goal is to provide access for users to configuration files via the CentralConfig, and give option for write, read and modify the settings.

Detector description
One of the principal parts in experimental physics software is that how the detector will be modeled and how will be the interface working to access instrument related data. This need to be separated by two important sections, to the Detector User Interface and the Detector Backend.

SHOE - The Shine Offline Event
The aim of the SHOE is to replace the legacy DS, T61 and NDST structures by one unified data format. It is based on ROOT and provides object oriented collection of classes for offline purposes. The content of information is scalable so supports different levels of detail.

Modules
The elements where users can insert unique processing elements inside the framework and manipulate the event structure, is the Module. The interface called VModule defines three virtual functions that users need to implement in the same method with the method return a ResultFlag in order to report success, failure or even instructions for the RunController that traces and monitors the work-flow of the offline application.

Support
The framework comes with many additional feature and support. Wide range of utilities from the fields of math, physics and computer science. Used in user modules (e.g. ODE Runge-Kutta Integrator) and also in the core framework (e.g. XML Reader, custom exceptions, shadow pointer). A tool called Shape to download and install every external dependencies with ease and to set up the required running environment for Shine. A buildbot was set up for automatization compilation, testing and validation after each commit. Doxygen documentation is generated in order to provide a clear picture about the framework and give information for Shine developers and power-users.

Achievements
- New, user friendly offline framework for the whole Collaboration.
- Stable and flexible system, with state-of-art techniques of computer science.
- Huge efforts to revive and integrate the legacy software within Shine.
- Collaboration members are migrating to the Shine Framework.

References

The framework comes with many additional feature and support. Wide range of utilities from the fields of math, physics and computer science. Used in user modules (e.g. ODE Runge-Kutta Integrator) and also in the core framework (e.g. XML Reader, custom exceptions, shadow pointer).

A tool called Shape to download and install every external dependencies with ease and to set up the required running environment for Shine. A buildbot was set up for automatization compilation, testing and validation after each commit. Doxygen documentation is generated in order to provide a clear picture about the framework and give information for Shine developers and power-users.

Achievements
- New, user friendly offline framework for the whole Collaboration.
- Stable and flexible system, with state-of-art techniques of computer science.
- Huge efforts to revive and integrate the legacy software within Shine.
- Collaboration members are migrating to the Shine Framework.

References