SoLID Simulation Requirements

Rakitha S. Beminiwattha*

Department of Physics, Syracuse University

(Dated: March 30, 2015)

* rakithab@jlab.org

I. INTRODUCTION

This is a summary of what SoLID experiment requires from a simulation software. Based on what is required, we can build a simulation framework using resources available. The simulation software framework can be divided into three sections, the input, the Simulation kernel and the output.

II. SIMULATION INPUT

Simulation input includes geometry, materials, event generators and other configurations including beam current, kinematics and other parameters. The inputs can easily be controlled by using macro files but a Graphical User Interface (GUI) to control inputs is preferred. See Figure 1.

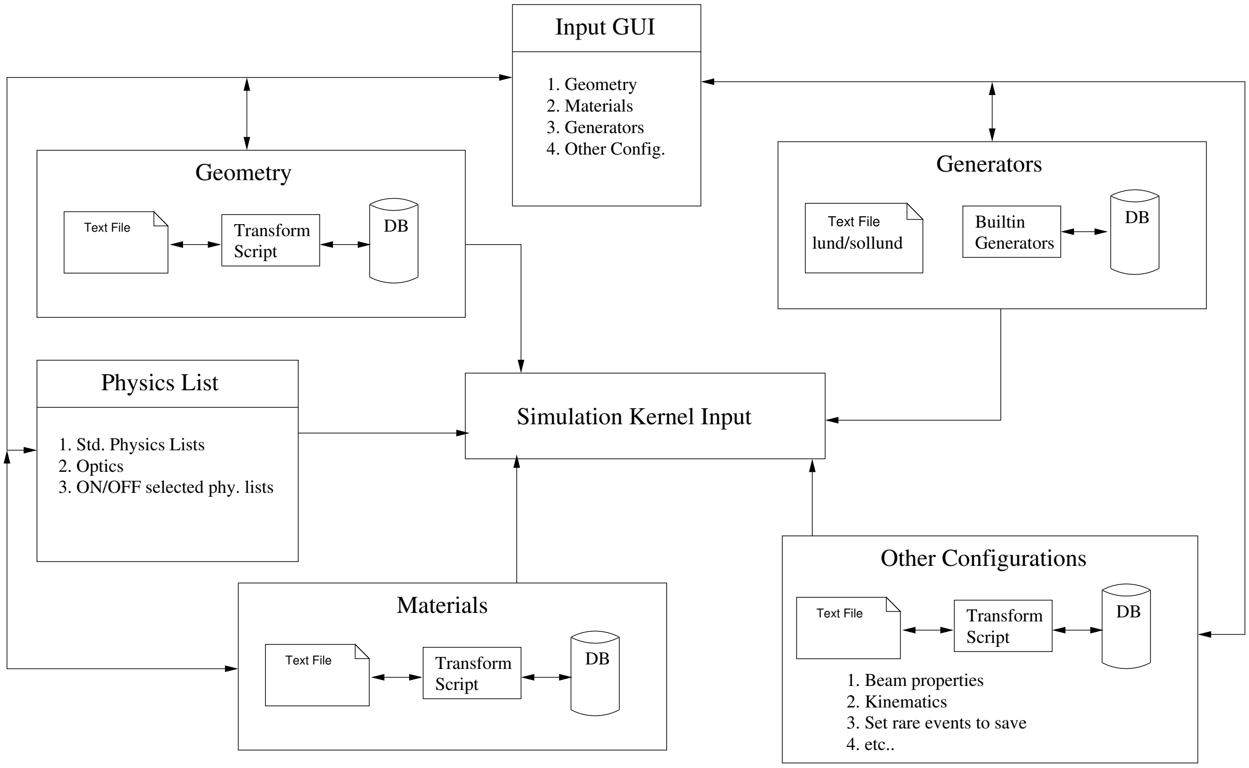

FIG. 1. Simulation input summary

A. Geometry

- Geometry can be read from a set of text files for local/development work
- Official/stable release geometry can be reads from a central or local database for simulations after development works
- Geometry parameters can be read either from a central OR local database or text file
  - Geometry text files can be generated from the database or database tables can be generated from the text files (using set of transform scripts)
- This can be a GEMC inspired design
- The database or text files are read by the simulation kernel to generate geometries using standard Geant4 geometry generation framework.
- It is not recommended to generate geometry through GDML
- Materials can be read from std. Geant4 tables (NIST and etc.) or text files or the database
- Need thorough volume overlap and overshoot checks.
- Need to have a mode to enable that will abort simulation if volume overlap and overshoot checks failed

B. Event Generator Related

- Can use built in generators or lund/sollund style input
- Support Built-in generators for important background or main signal simulations (Pions, DIS, elastic, inelastic, ..) - Can be imported from Remoll
- Better to have number weighted generators that will save computing time and will be useful for trigger and background related simulations
- Remoll has number weighted pions background generators

C. Other Configurations

- Define output format
- Kinematics
- Define rare event tracking
- Raster and other electron beam configurations
- Digitization parameters and channel maps and etc.
- ....

D. Input GUI

- Have a GEMC style GUI to control all the simulation inputs

III. SIMULATION CORE SYSTEM (KERNEL)

The core of SoLID simulation can be developed using widely available Geant4 code framework or a forked version of the GEMC 2.0 framework. A summary of basic requirements of a simulation kernel that should suite SoLID simulation are listed here.

- Directly use Geant4 simulation framework with C++ and Root libraries
- Can be build using either GEMC framework or based on standard C++ and Geant4 framework based (Remoll style)
- Needs to support kryptonite materials and proper energy deposit support for it -
- Needs to save random number generator and input status for less time consuming Geant4 particle tracking
- Save output to root directly
- Have the choice to save to evio format that will be helpful for future main analyzer input and trigger simulation input

- Ability to turn ON/OFF individual physics processes : for example, enable only EM or Hadronic physics
- Optical physics support
- Digitization : This could be part of simulation output but good to be built into the framework

IV. SIMULATION OUTPUT

A summary of required simulation output is given here. See attached figure 2 for a diagram of simulation output.

- Self-explanatory output : Ability to access all the input parameters or links to access input parameters, the goal is to be able to reproduce simulation to get same output

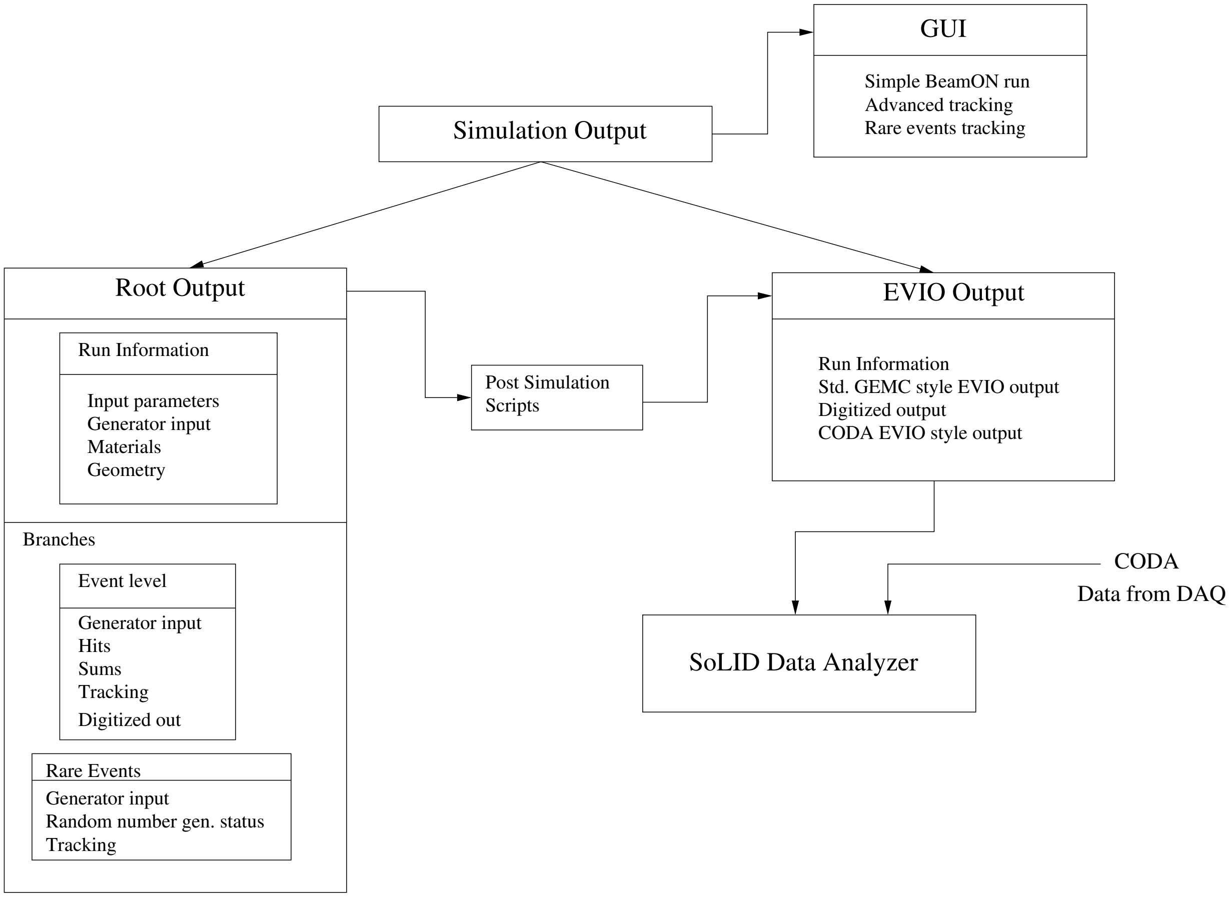

FIG. 2. Simulation output summary

- Main output will be root
- provide EVIO support mainly for replicating CODA style output for main SoLID data analyzer and trigger simulation
- GUI output for tracking : simple beamOn tracking and more advanced rare event tracking
  - Can import GEMC GUI design with added above features