Geometry Database — Design Ideas

1. Describe geometry by sets of human-friendly core parameters. Simulation, digitization, reconstruction and analysis each build their internal geometry structures from these core parameters
   (e.g. "sensitive" vs. "logical" volumes)
2. Ensure consistency between different stages (sim, digi etc.) Most easily achieved by storing actually used parameters in output files
3. Central vs. local database. Local should be user-friendly (no servers). For digi/reco/analysis, input file would be one of the database sources.
4. Allow easy override of parameters (but maintain consistency)
5. Support version control of parameter sets
6. Provide user-friendly visualization and editing of parameters
7. Should be compatible with other JLab efforts. Must work with GEMC

Considerations (I)

- Item 1: Develop generic API for accessing core parameters. Should support various backends (input file, MySQL, `sqlite`, CCDB, XML, flat text files, `git`, etc.) SoLID will implement only its own preferred set of backends.
- Item 1: Develop parameterizations for all geometries. Some already done in GEMC's scripts, go from there.
- Item 1: Parameters will evolve, hence need robust schema evolution. ROOT, text, XML (?), CCDB (?) support this.
- Item 2: Ideally independent of file format $\longrightarrow$ binary formats (e.g. ROOT) not very suitable, prefer text

Considerations (II)

- Item 4: Maintaining consistency could be tricky. Need to stop users from overriding parameters of previous stages.
- Item 5: CCDB offers some version control. Text/XML files under `git` would be even better.
- Item 6: Writing a GUI editor is a lot of work
- Item 7: Integration into GEMC is an issue. Would need to modify geometry processing (?) and output. But GEMC development is becoming more flexible. Strongly prefer not to fork GEMC.