New CAD / Mirrors Import Mechanisms in GEMC

M. Ungaro

*Jefferson Lab, 12000 Jefferson Avenue, 23606 Newport News, VA, USA*

**Abstract**

The CAD import mechanism in GEMC has been generalized to account for geometry variations. A new CADSQLITE factory is introduced to allow for Run Number indexing of CAD imported geometry, with definitions stored in a SQLITE database. In addition, the SQLITE factory has been extended to define and store mirrors properties.

**Overview**

To import volumes from CAD, before version 5.12, GEMC simulations used strings in the steering cards (gcards) to specify directories containing the CAD files. An optional JSON file could be given to specify additional attributes of the volumes such as displacements, material, mother volume. This approach had several drawbacks:

* It cannot be generalized and used for other geometry factories like SQLITE
* Geometry variations were not properly accounted for; to go around this limitation for CLAS12 we have been copying directories to different path names, each with its own JSON file specifying the variation changes. For example:

<detector name="ctof/javacad\_rga\_spring2018/" factory="CAD"/>

<detector name="ctof/javacad\_rga\_fall2018/" factory="CAD"/>

This ugly workaround duplicates files and is eliminated in this work.

* All files in the directory were loaded indiscriminately, even those unwanted for some variations; to go around this limitation additional attributes have been used in the steering card to mark volumes as non-existent for configurations that didn’t have them. For example:

<detector name="FMT">

<existence exist="no" />

</detector>

This was an ugly workaround and is eliminated in this work.

To provide a simpler, cleaner, general mechanism, the following changes have been implemented:

* Geometry variations have been implemented using the JSON filenames
* Only volumes specified in the JSON files are loaded
* SQLITE support has been added.

This will eliminate the need of duplicating directories and using additional attributes in the steering cards.

In addition, SQLITE support has been extended to define and store mirrors properties.

**API Changes: CAD factory in the steering card**

The usual CAD factory mechanism has been modified to include variations. Entries in the steering cards such as:

<detector name="htcc/cad\_rga\_fall2018/" factory="CAD"/>

Have been replaced with:

<detector name="htcc/cad/" factory="CAD" variation="rga\_fall2018"/>

This will load the gxml htcc/cad/cad\_ fall2018.gxml that contains the list of volumes to be loaded. If a volume is not wanted for that variation, its absence in the gxml file will tell GEMC not to load it.

Notice the same htcc/cad/subdir is shared among all variations and no duplicate directories or additional gcard attributes are needed in the gcards.

**SQLITE factory for CAD volumes**

New code in the api/perl/sqlite.py utility has been added to create a sqlite database table ‘cad’ with all the modifiers supported in the JSON api. A new PERL api has been added to fill the cad table with the necessary entries. An example code to import an HTCC CAD volume is:

my %cad = init\_cad();

$cad{"name"} = "htccMollerConeExt";

$cad{"color"} = "888888";

$cad{"material"} = "rohacell31";

$cad{"position"} = "0\*cm 0\*cm $zpos1\*cm";

$cad{"rotation"} = "0\*deg 180\*deg 0\*deg";

print\_cad(\%configuration, \%cad);

The cad api fills the table with the relevant entries, see Figure 1. The system name has “\_cad” appended to the name to keep the uniqueness of detector names in the gcards. Consequentially entries such as:

<detector name="htcc" factory="TEXT" variation="rga\_spring2018"/><detector name="cad/" factory="CAD" variation="rga\_spring2018"/>

Using the SQLITE factories are written as:

<detector name="htcc" factory="SQLITE"/>  
<detector name="htcc\_cad" factory="SQLITECAD"/>

Notice the variation is “default” by default, and it’s not mentioned (but it could be) in the SQLITE factories because run number is used instead.

As an example, in Fig. 1 the HTCC CAD definitions are shown for the original (run=11), rga\_spring2018 (run=3029) and rga\_fall2018 (run= 4763) configurations.

A picture containing graphical user interface

Description automatically generated

*Figure 1: the HTCC CAD geometry definitions in the SQLITE database for three run numbers.*

A new function in the utility file clas12\_runs\_and\_variations.pm has been added to return the experiment configuration given a variation or run number input. This will ensure code is not duplicated when having to change parameters based on a variation or a run number in the perl scripts.

**API Changes: API PERL SQLITE factory for Mirrors**

New code in the api/perl/sqlite.py utility has been added to create a sqlite database table ‘mirrors’ containing the properties needed to define reflectivity, refraction index, finish type, etc. The PERL api for mirrors has been extended to fill the mirrors table with the same entries as the TEXT api. The same code used to define mirror surfaces is also used to fill the SQLITE database, making the transition transparent to users. The output results in entries in the ‘mirrors’ table when the SQLITE factory is selected. An example of usch entries is shown in Figure 2 for the HTCC system.

**Graphical user interface, application

Description automatically generated**

*Figure 2: the HTCC mirrors definitions in the SQLITE database for three run numbers.*

**CLAS12 CAD imports changes**

Below are the list of changes to gemc/detectors/clas12 that use the new CAD import mechanism:

* **HTCC**:
* cad/cad.gxml 🡪 cad/cad\_original.gxml
* cad\_spring18/cad.gxml 🡪 cad/cad\_rga\_spring2018.gxml
* cad\_fall/cad.gxml 🡪 cad/cad\_rga\_fall2018.gxml
* ltcc/cad\_cone/cad.gxml
* ltcc/cad/cad.gxml
* ddvcs/cad/cad.gxml
* targets/hdIce/cad.gxml
* targets/rge-dt/Pb/cad.gxml
* targets/rge-dt/Empty/cad.gxml
* targets/rge-dt/Al/cad.gxml
* targets/rge-dt/Cu/cad.gxml
* targets/rge-dt/common/cad.gxml
* targets/rge-dt/Sn/cad.gxml
* targets/rge-dt/C/cad.gxml
* targets/ltarget/cad.gxml
* targets/transverseTargetCad/cad.gxml
* targets/cade/cad.gxml
* targets/cad/cad.gxml
* targets/cadrgm/cad.gxml
* targets/flagCad/cad.gxml
* targets/PolTarg/cad.gxml
* ctof/javacad\_default\_upstream/cad.gxml
* ctof/javacad\_rga\_fall2018\_upstream/cad.gxml
* ctof/javacad\_rga\_spring2018/cad.gxml
* ctof/javacad\_rga\_fall2018/cad.gxml
* ctof/javacad\_rga\_spring2018\_upstream/cad.gxml
* ctof/javacad\_default/cad.gxml
* magnets/cad/cad.gxml
* rich/cad\_default/cad.gxml
* rich/cad\_rgc\_summer2022/cad.gxml
* rich/cad\_rga\_fall2018/cad.gxml
* nuclearTargetTest/cad/cad.gxml
* beamline/cadBeamline/cad.gxml
* beamline/cad\_downstream\_beamline/cad.gxml
* beamline/cadBeamlineELMO/cad.gxml
* beamline/cadBeamlineFTOFF/cad.gxml
* beamline/transverseTargetMagnets/forwardWedges/cad.gxml
* beamline/transverseTargetMagnets/midpointWedges/cad.gxml
* beamline/transverseTargetMagnets/noWedges/cad.gxml

**Gcards changes in clas12-config**

Below are the changes to the GCARDS in the clas12-config repository. The proper variations replace the cad subdir paths directives. Many variations definition replace GCARDS entries to displace or remove detector elements.

In addition, the “default” string used when specifying detector variation is removed as it is used by default in GEMC.

**HTCC**

The text “original” variation has been replaced with rga\_spring2018 and rga\_fall2018 when applicable. In addition, the entries:

<detector name="htcc/javacad\_rga\_spring2018/" factory="CAD"/>

<detector name="htcc/javacad\_rga\_fall2018/" factory="CAD"/>

Have been replaced with, respectively:

<detector name="htcc/cad/" factory="CAD" variation="rga\_spring2018"/>

<detector name="htcc/cad/" factory="CAD" variation="rga\_fall2018"/>

The HTCC displacements in all gcards have been removed as they are incorporated in the variations.

**Summary**

**Conclusions**

Both the rates and the radiation damage benefit from 51 microns of tungsten shield around the CLAS12 scattering chambers. There is no benefit in additional thicknesses. The rates have been compared with physics run data at several beam currents. There is a good agreement between the real and the simulated data.

**References**

[1] *M. Ungaro*, clas12 simulation software / geometry tags: https://github.com/gemc/clas12Tags.