

SiD in DDSim

Dan PROTOPOPESCU - GLASGOW, UK

April 2016

SiD @ Glasgow

The **Glasgow Linear Collider Group** is working on the implementation of the SiD model in DD4hep, and is aiming to deliver a physics-ready model in close collaboration with SiD sub-detector experts and DD4hep developers.

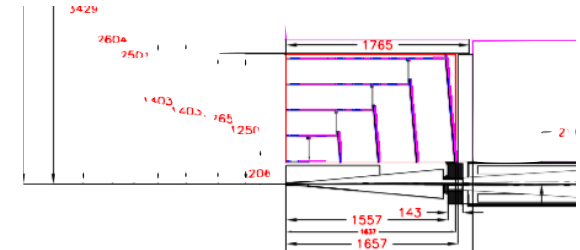
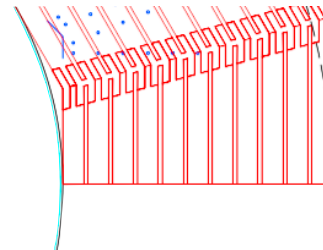
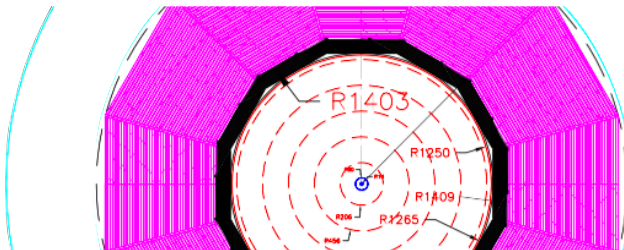
We are: A. Robson, **D. Protopopescu** and B. Mischenko (MSc student)

Our work involves

- 1) verifying the SiD geometry ported from `slcic`
- 2) updating the code to the latest SiD model and parameters
- 3) validating the new detector implementation
- 4) maintaining software and configuration compatibility with the latest DDRec/DDSim development versions
- 5) providing the validated model to the detector optimisation group

Along the way we

- Incorporate detector geometry and spec changes as the SiD model evolves.
- Document every step of the process for other users/developers within the SiD and the wider LC communities.
- Maintain contact with ILC, CLIC and ILD developers.



Obtaining the code

At the moment, our code is committed to the DESY SVN repository, where the core DD4hep software and the CLIC and ILD models are stored as well. This is to make it easier to include SiD in the standard tests.

The latest/work version is labelled **SiD_o1_v01**. The XML configuration is located in `lcgeo/SiD/compact/SiD_o1_v01` and the C++ drivers are located in `lcgeo/detector/`

Browse or check out the code via <https://svnsrv.desy.de/websvn/wsvn/General.ddsim/lcgeo>

Build example

```
svn co https://svnsrv.desy.de/basic/ddsim/lcgeo/trunk lcgeo

source /cvmfs/sft.cern.ch/lcg/releases/gcc/4.8.4/x86_64-slc6/setup.sh
source /cvmfs/ilc.desy.de/sw/x86_64_gcc48_sl6/v01-17-09/init_ilcsoft.sh

cd lcgeo/
mkdir build
cd build
cmake -DCMAKE_CXX_COMPILER=`which g++` -DCMAKE_C_COMPILER=`which gcc` -C /cvmfs/ilc.desy.de/sw/x86_64_gcc48_sl6/ILCSoft.cmake ..
make -j4
make install
source ../bin/thislcgeo.sh
```

Visualising the model

Build and display detector model with

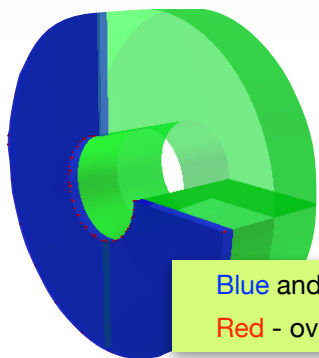
```
cd lcgeo/  
source bin/thislcgeo.sh  
geoDisplay SiD/compact/SiD_o1_v01/SiD_o1_v01.xml
```

or

```
teveDisplay SiD/compact/SiD_o1_v01/SiD_o1_v01.xml
```

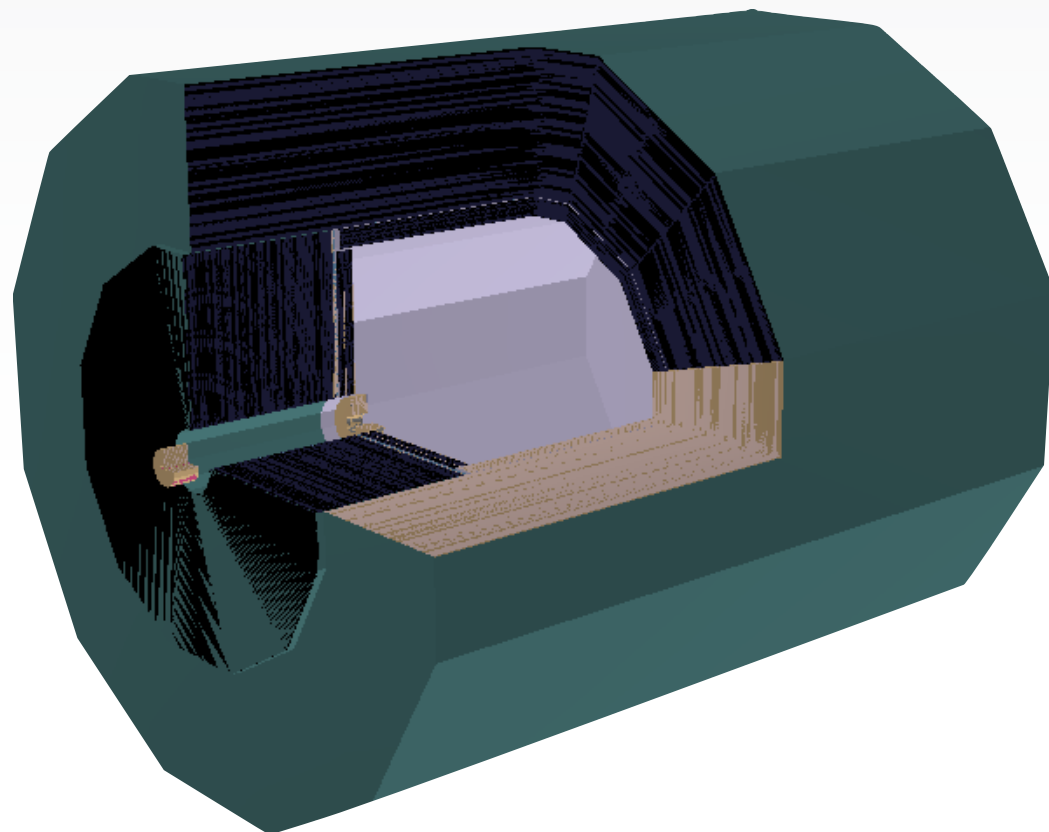
Check overlaps with

```
gGeoManager->CheckOverlaps(0.01);  
gGeoManager->PrintOverlaps();
```



Blue and green - the two overlapping volumes
Red - overlap points

Example overlap in LumiCal



SiD_o1_v01

All sub-detectors are built at the moment using generic DD4hep drivers. Generic drivers are best for testing DDSim. Once testing is done, we will carry over the final version of the extensions to the detailed drivers.

SiD Subcomponent	XML	Driver
Vertex Barrel	sidloi3/in work	DD4hep_SiTrackerBarrel
Vertex Endcap	sidloi3/in work	DD4hep_SiTrackerEndcap2
Tracker Barrel	sidloi3/in work	DD4hep_SiTrackerBarrel
Tracker Endcap	sidloi3/in work	DD4hep_SiTrackerEndcap2
Ecal Barrel	EcalBarrel_o1_v01_00	GGenericCalBarrel_o1_v01
Ecal Endcap	EcalEndcap_o1_v01_00	GGenericCalEndcap_o1_v01
Hcal Barrel	HcalBarrel_o1_v01_00	GGenericCalBarrel_o1_v01
Hcal Endcap	HcalEndcap_o1_v01_00	GGenericCalEndcap_o1_v01
Muon Barrel	sidloi3/in work	DD4hep_PolyhedraBarrelCalorimeter2
Muon Endcap	sidloi3/in work	DD4hep_PolyhedraEndcapCalorimeter2
LumiCal	LumiCal_o1_v01_00	DD4hep_CylindricalEndcapCalorimeter
BeamCal	BeamCal_o1_v01_00	DD4hep_ForwardDetector
Beam pipe	sidloi3/in work	DD4hep_PolyconeSupport/TubeSegment
Supports and readout	sidloi3/in work	none

■ slic port, ■ updated XML, ■ DD4hep generic drivers, ■ Generic ILD/CLIC/SiD drivers that include Pandora extensions

SiD_o1_v01

All sub-detectors are built at the moment using generic DD4hep drivers. Generic drivers are best for testing DDSim. Once testing is done, we will carry over the final version of the extensions to the detailed drivers.

SiD Subcomponent	XML	Driver
Vertex Barrel	sidloi3/in work	DD4hep_SiTrackerBarrel
Vertex Endcap		DD4hep_SiTrackerEndcap2
Tracker Barrel		DD4hep_SiTrackerBarrel
Tracker Endcap		DD4hep_SiTrackerEndcap2
ECal Barrel		GGenericCalBarrel_o1_v01
ECal Endcap		GGenericCalEndcap_o1_v01
HCal Barrel		GGenericCalBarrel_o1_v01
HCal Endcap		GGenericCalEndcap_o1_v01
Muon Barrel		DD4hep_PolyhedraBarrelCalorimeter2
Muon Endcap	sidloi3/in work	DD4hep_PolyhedraEndcapCalorimeter2
LumiCal	LumiCal_o1_v01_00	DD4hep_CylindricalEndcapCalorimeter
BeamCal	BeamCal_o1_v01_00	DD4hep_ForwardDetector
Beam pipe	sidloi3/in work	DD4hep_PolyconeSupport/TubeSegment
Supports and readout	sidloi3/in work	none

Work done:

- Updated XMLs to latest naming and structure conventions
- Updated dimensions as per latest engineering drawing
- Moved readouts and plugins to the individual detector descriptions to assure modularity
- Verified visualisation
- Checked overlaps and extrusions
- Code streamlining

■ slic port, ■ updated XML, ■ DD4hep generic drivers, ■ Generic ILD/CLIC/SiD drivers that include Pandora extensions

SiD_o1_v01

All sub-detectors are built at the moment using generic DD4hep drivers. Generic drivers are best for testing DDSim. Once testing is done, we will carry over the final version of the extensions to the detailed drivers.

	SiD Subcomponent	XML	Driver
need input from sub-detector experts	Vertex Barrel	sidloi3/in work	DD4hep_SiTrackerBarrel
	Vertex Endcap	sidloi3/in work	DD4hep_SiTrackerEndcap2
	Tracker Barrel	<ul style="list-style-type: none"> Tracker barrel can be greatly simplified; same for LumiCal, BeamCal. see work done by Marko on the CLIC trackers ... 	DD4hep_SiTrackerBarrel
	Tracker Endcap		DD4hep_SiTrackerEndcap2
	ECal Barrel		GGenericCalBarrel_o1_v01
	ECal Endcap		GGenericCalEndcap_o1_v01
	HCal Barrel	HcalBarrel_o1_v01_00	GGenericCalBarrel_o1_v01
	HCal Endcap	HcalEndcap_o1_v01_00	GGenericCalEndcap_o1_v01
	Muon Barrel	sidloi3/in work	DD4hep_PolyhedraBarrelCalorimeter2
	Muon Endcap	sidloi3/in work	DD4hep_PolyhedraEndcapCalorimeter2
	LumiCal	LumiCal_o1_v01_00	DD4hep_CylindricalEndcapCalorimeter
	BeamCal	BeamCal_o1_v01_00	DD4hep_ForwardDetector
	Beam pipe	sidloi3/in work	DD4hep_PolyconeSupport/TubeSegment
	Supports and readout	sidloi3/in work	none

■ slic port, ■ updated XML, ■ DD4hep generic drivers, ■ Generic ILD/CLIC/SiD drivers that include Pandora extensions

Running the simulation

Might need to set

```
export LCGRELEASES=/cvmfs/sft.cern.ch/lcg/releases/LCG_84
export PYTHONDIR=$LCGRELEASES/Python/2.7.10/x86_64-slc6-gcc48-opt
export PATH=$PYTHONDIR/bin:$PATH
export LD_LIBRARY_PATH=$PYTHONDIR/lib:$LD_LIBRARY_PATH
```

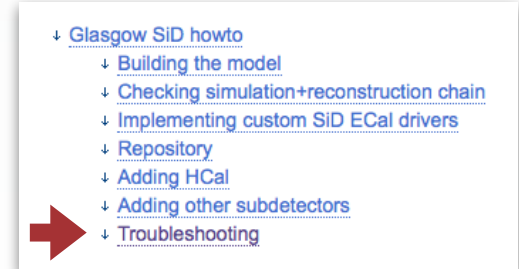
also

```
export LCGRELEASES=/cvmfs/sft.cern.ch/lcg/releases/LCG_84
export PYTHONDIR=$LCGRELEASES/Python/2.7.10/x86_64-slc6-gcc48-opt
export PATH=$PYTHONDIR/bin:$PATH
export LD_LIBRARY_PATH=$PYTHONDIR/lib:$LD_LIBRARY_PATH
export PYTOOLSDIR=$LCGRELEASES/pytools/1.9_python2.7/x86_64-slc6-gcc48-opt
export PYTHONPATH=$PYTOOLSDIR/lib/python2.7/site-packages:$PYTHONPATH
export PATH=$PYTOOLSDIR/bin:$PATH
[ or try source /cvmfs/ilc.desy.de/sw/x86_64_gcc48_sl6/init_gcc48.sh ]
```

Try for example

- 1 `make test` [might not work due to gun options mismatch]
`cd ..`
- 2 `python example/lcio_particle_gun.py`
`ddsim --compactFile=SiD/compact/SiD_o1_v01/SiD_o1_v01.xml --runType=batch --inputFile mcparticles.slcio -N=1 --`
`outputFile=testSiD_o1_v01.slcio`
- 3 `materialScan SiD/compact/SiD_o1_v01/SiD_o1_v01.xml 0 0 0 100 0 0`

Visit the troubleshooting section or email Dan if problems ...



Summary

Done:

- slic sidloi3 port using generic drivers (F. Gaede)
- plugins and readouts moved to the individual detector's XML to assure modularity
- updated XMLs to the latest variable structure and naming conventions; updated dimensions

In progress by us:

- more work on XMLs and drivers
- committing code and writing documentation
- focus on simulation

To do (in general):

- finalise DD4hep implementation of all SiD sub-detectors
- customise C++ drivers to include more detailed geometry
- provide code for physics simulations and optimisation studies
- **benchmark reconstruction** (new MSc student)
- update everything if/when software and/or geometry are modified



The screenshot shows the University of Glasgow Experimental Particle Physics website. The page title is 'Glasgow SiD howto'. The content includes a navigation menu on the left with sections like 'My links', 'LinearCollider Web', and 'Webs'. The main content area has a breadcrumb trail 'TWiki > LinearCollider Web > GlaSiD (2016-04-25, DanProtopopescu)'. Below this is the title 'Glasgow SiD howto' and a paragraph: 'The starting point is the `sidloi3` model in `lcgeo` implemented by F. Gaede and based on the original muons and reconstructing them (see `lcgeo/example/run_sid_reco.xml`).'. A list of links follows: 'Glasgow SiD howto', 'Building the model', 'Checking simulation+reconstruction chain', 'Implementing custom SiD ECal drivers', 'Repository', 'Adding HCal', 'Adding other subdetectors', and 'Troubleshooting'. Below the links is a section titled 'Building the model' with text: 'The model is using the generic detector drivers in DD4hep. For the calorimeters (Ecal/Hcal) the current DDRec extensions that will eventually be needed by DDMarlinPandora. To compile, from the local `lcgeo/` directory:'. A 3D model of a detector is shown in the background, partially obscured by a green box.

Student's task:

start running the **DDR**ec reconstruction chain and evaluate its readiness for SiD

THANKS!

QUESTIONS ?
SUGGESTIONS ?

Visit <https://twiki.ppe.gla.ac.uk/bin/view/LinearCollider/GlaSiD>

D.Protopopescu - April 2016