



ПРОГРАММНАЯ ПЛАТФОРМА ДЛЯ МНОГОЭТАПНОЙ ОБРАБОТКИ ДАННЫХ ЭКСПЕРИМЕНТА SPD

Симбирятин Л. Л.

Научный руководитель: Солдатов Е. Ю. (каф. №40, МИФИ)

Научный консультант: Жемчугов А. С. (ЛЯП, ОИЯИ)

25.06.2025

Москва

Цели работы

Текущее оффлайн ПО SPDRoot:

- недостаточно надежное, много рудиментарного кода
- наследует проблемы пакета ROOT
- однопоточное

Таким образом, его необходимо заменить.

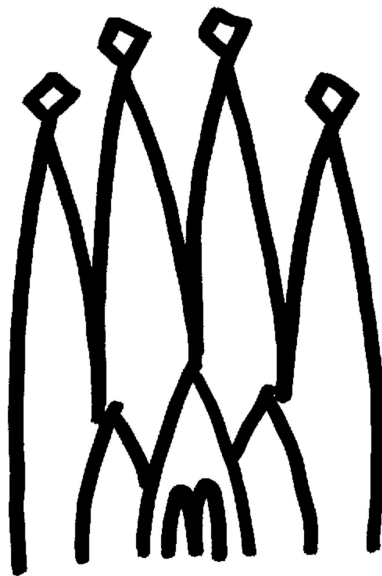
Gaudi - фреймворк в качестве альтернативы:

- надежный
- динамически конфигурируемые задачи
- поддержка многопоточности

Gaudi Tutorial: <https://git.jinr.ru/lsimbir/gauditutorial/-/tree/main>

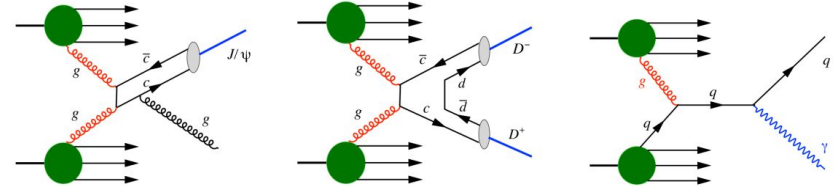
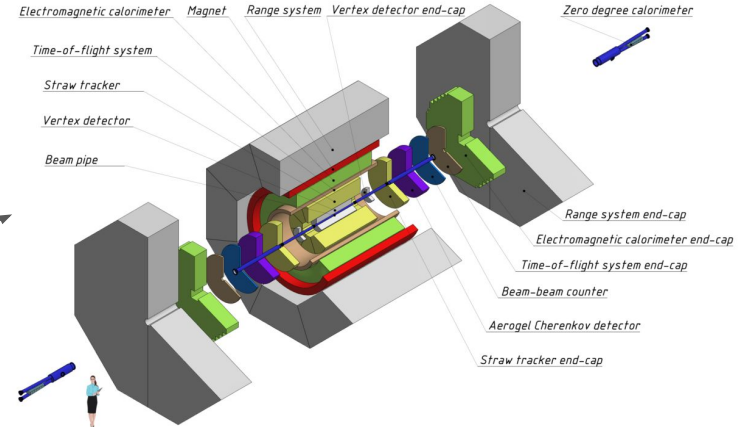
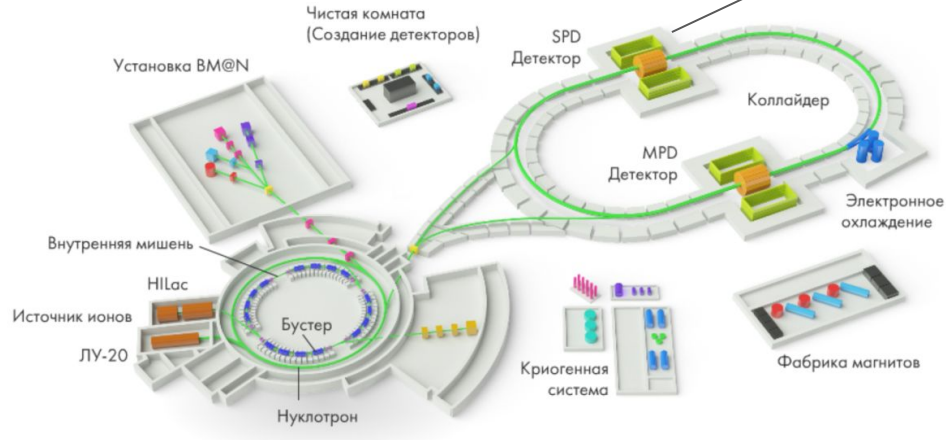
Gaudi Docker image: <https://git.jinr.ru/spd/spd-sw/gaudi>

Sampo repo: <https://git.jinr.ru/spd/spd-sw/sampo/>

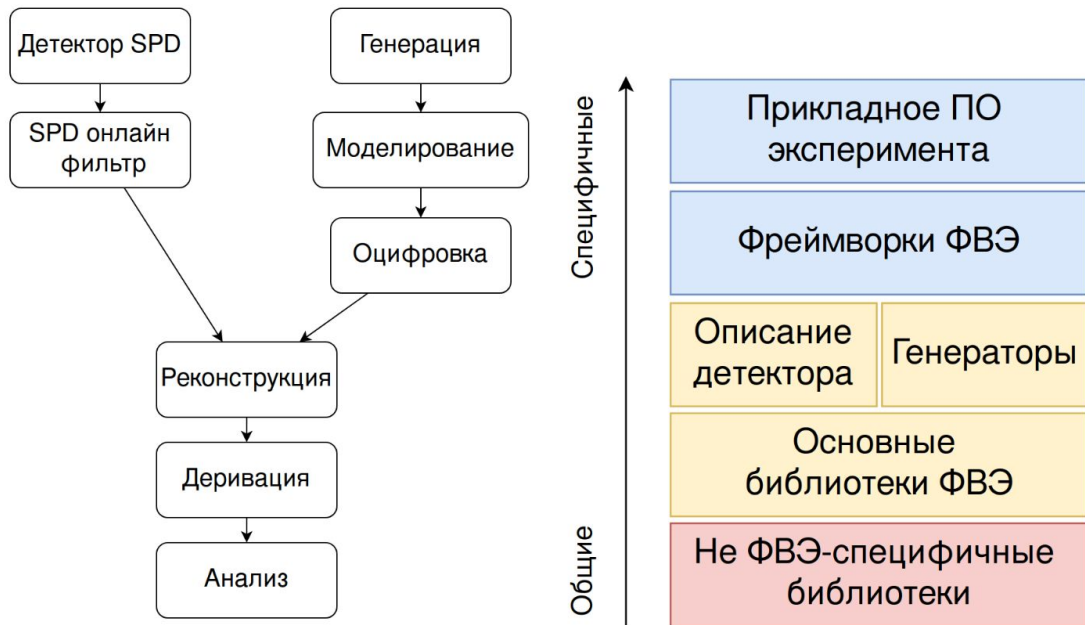


Эксперимент SPD

Основная цель эксперимента SPD - извлечь информацию о глюонных функциях распределения, зависящих от поперечного импульса (TMD PDFs), в поляризованных pp ($\sqrt{s} = 27$ ГэВ) и dd ($\sqrt{s} = 13.5$ ГэВ) столкновениях. Ожидаемая светимость – $10^{32} \text{см}^{-2} \text{с}^{-1}$.



Программное обеспечение в ФВЭ



Pythia8 - генерация первичных вершин

HepMC3 - универсальный формат события, IO

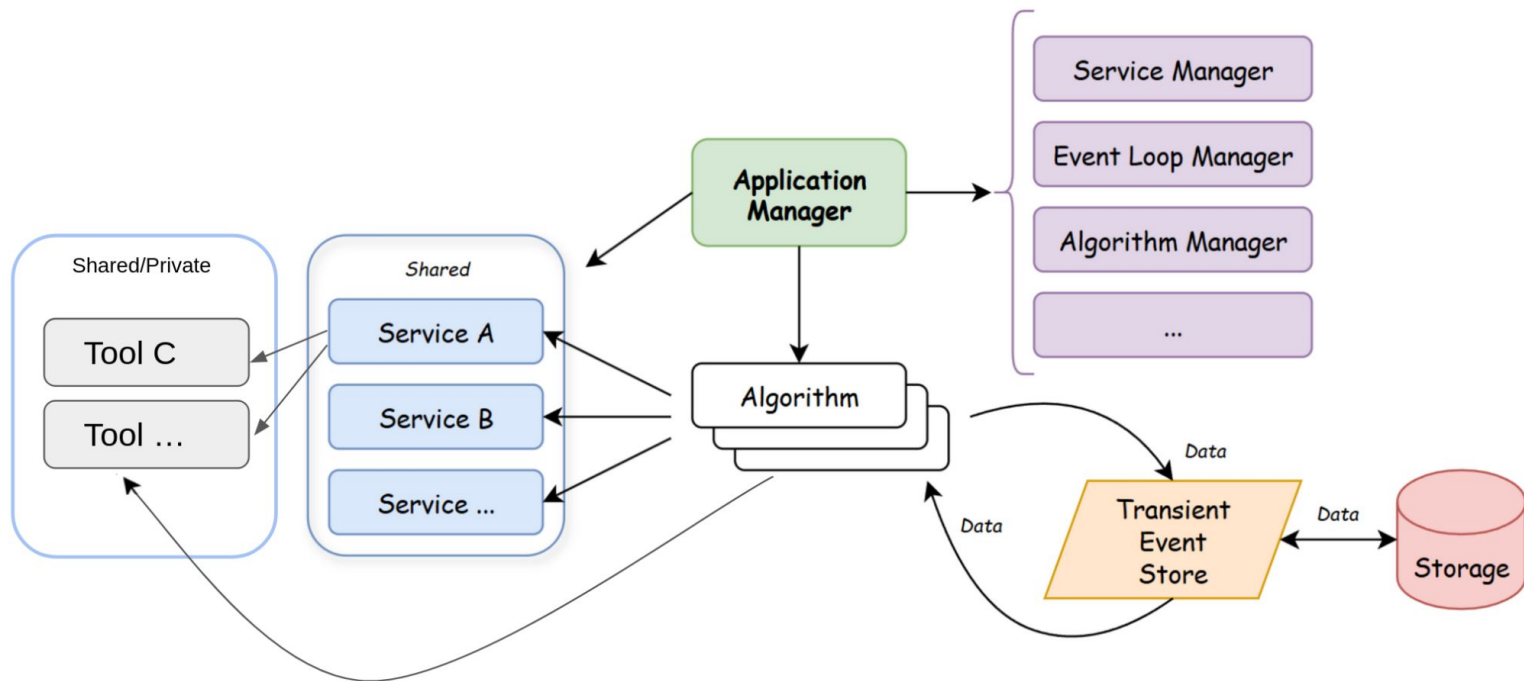
GeoModel - описание детектора, чтение из БД

Geant4 - моделирование взаимодействия частиц с веществом детектора



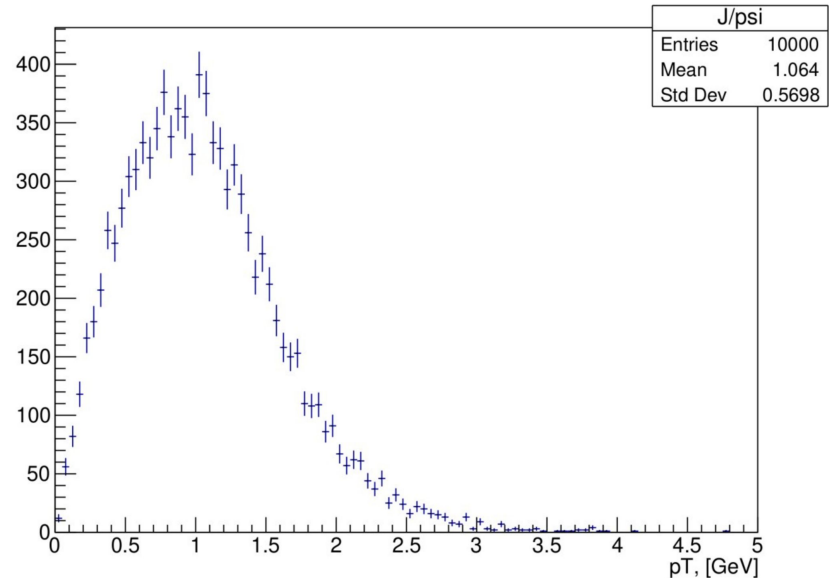
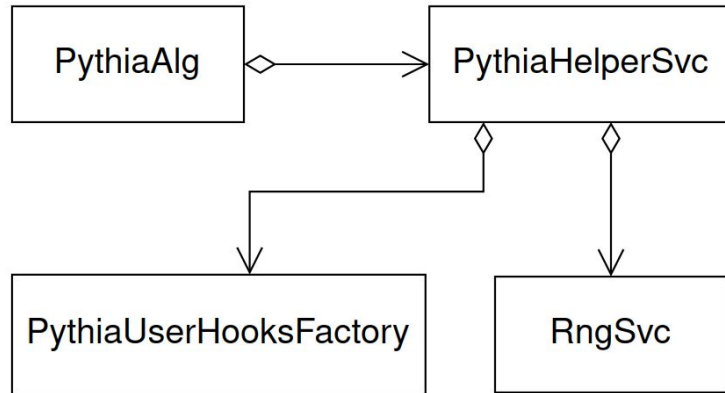
Архитектура Gaudi (однопоточная)

Концепция: Алгоритмы запускаются в цикле обработки событий и совершают операции над объектами данных. Часть работы они делегируют сервисам и инструментам (Tools). Все компоненты динамически конфигурируемые через JobOptions.



Pythia8

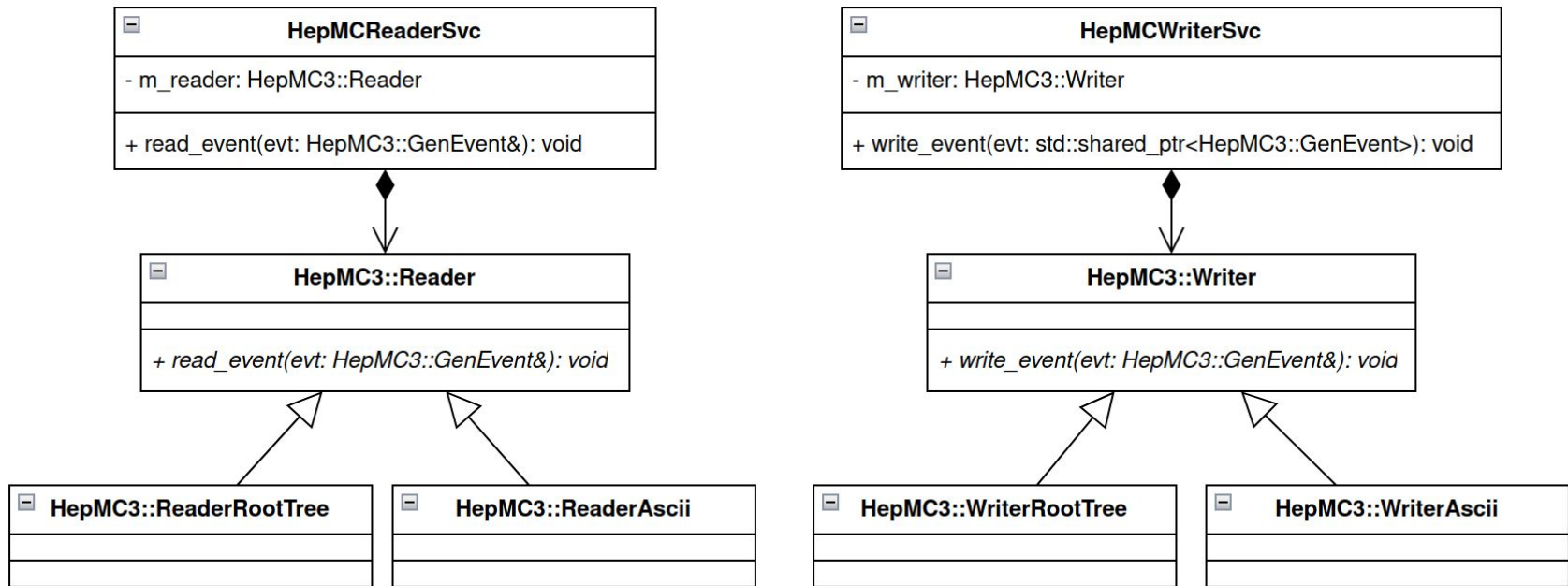
Концепция: PythiaHelperSvc создает объекты класса Pythia8, снабжает их генераторами случайных чисел, юзер-хуками. Pythia8::next() вызывается в алгоритме PythiaAlg.



pp, $\sqrt{s} = 27 \text{ ГэВ}$

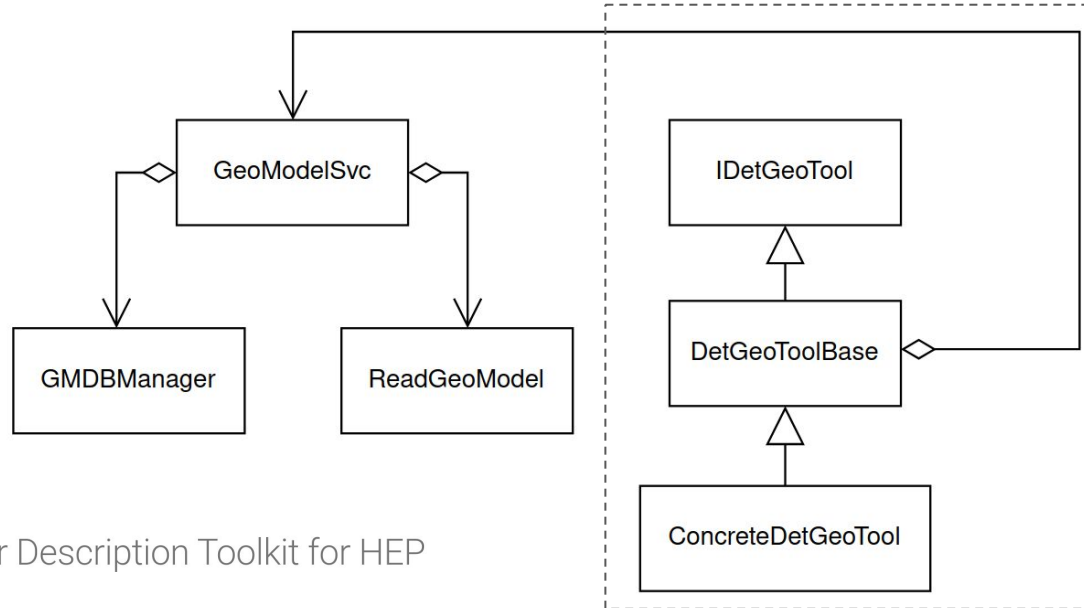
НepMC3 IO

Концепция: HepMCReaderSvc и HepMCWriterSvc осуществляют IO в формате HepMC3. HepMCReaderSvc читает последовательно все входные файлы, а прочитав последнее событие планирует остановку EventLoop.



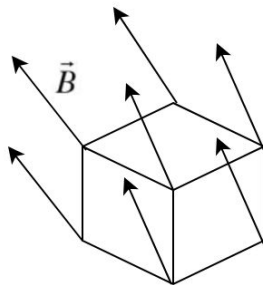
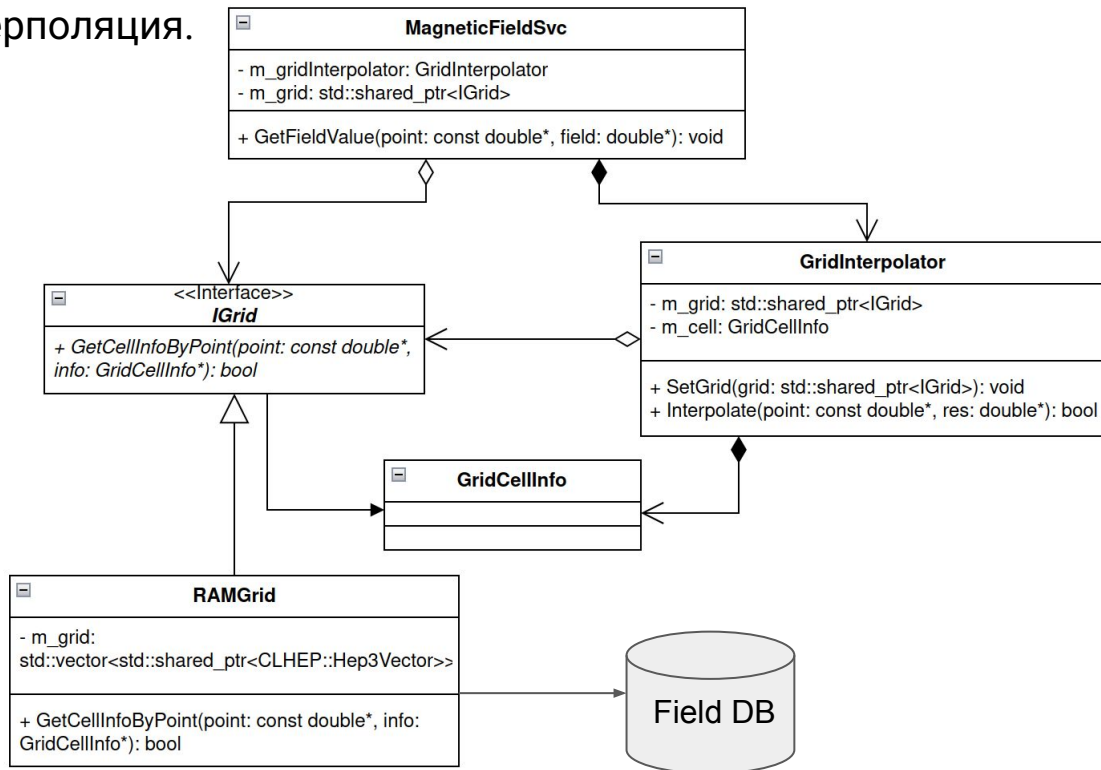
GeoModel

Концепция: GeoModelSvc имеет доступ к БД с геометрией через GMDBManager. ReadGeoModel используется для построения геометрии. IDetGeoTools являются клиентами сервиса, инициализируются и содержат информацию о подсистемах детектора.

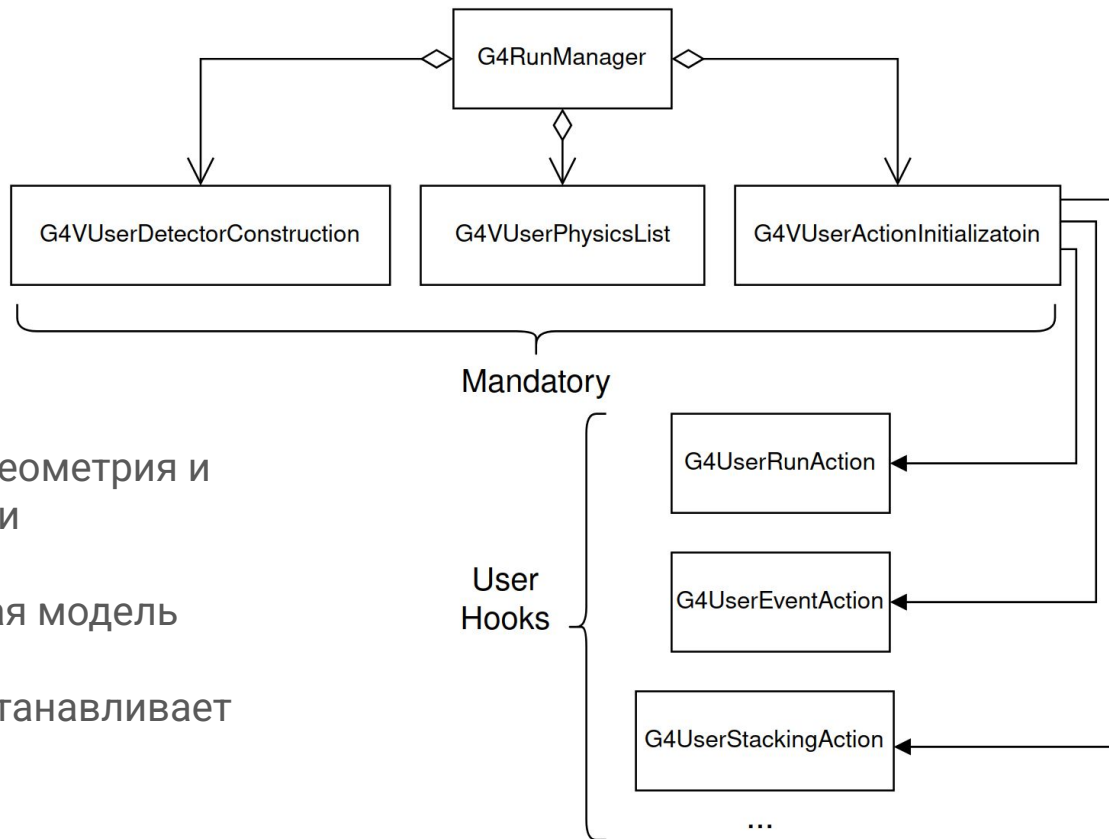


Магнитное поле

Концепция: Карта поля хранится в SQLite БД. Для расчетов используется трилинейная интерполяция.



Geant4: Пользовательские классы



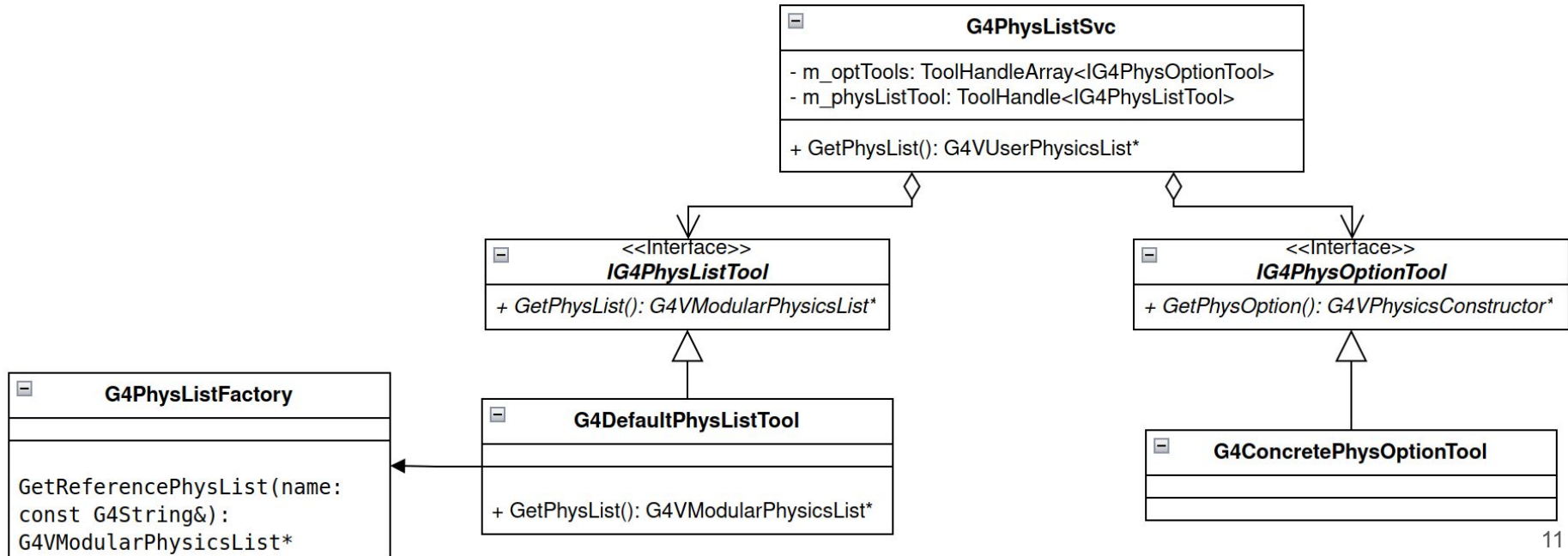
G4VUserDetectorConstruction -> геометрия и чувствительные части установки

G4VUserPhysicsList -> физическая модель

G4VUserActionInitioalization -> устанавливает пользовательские хуки

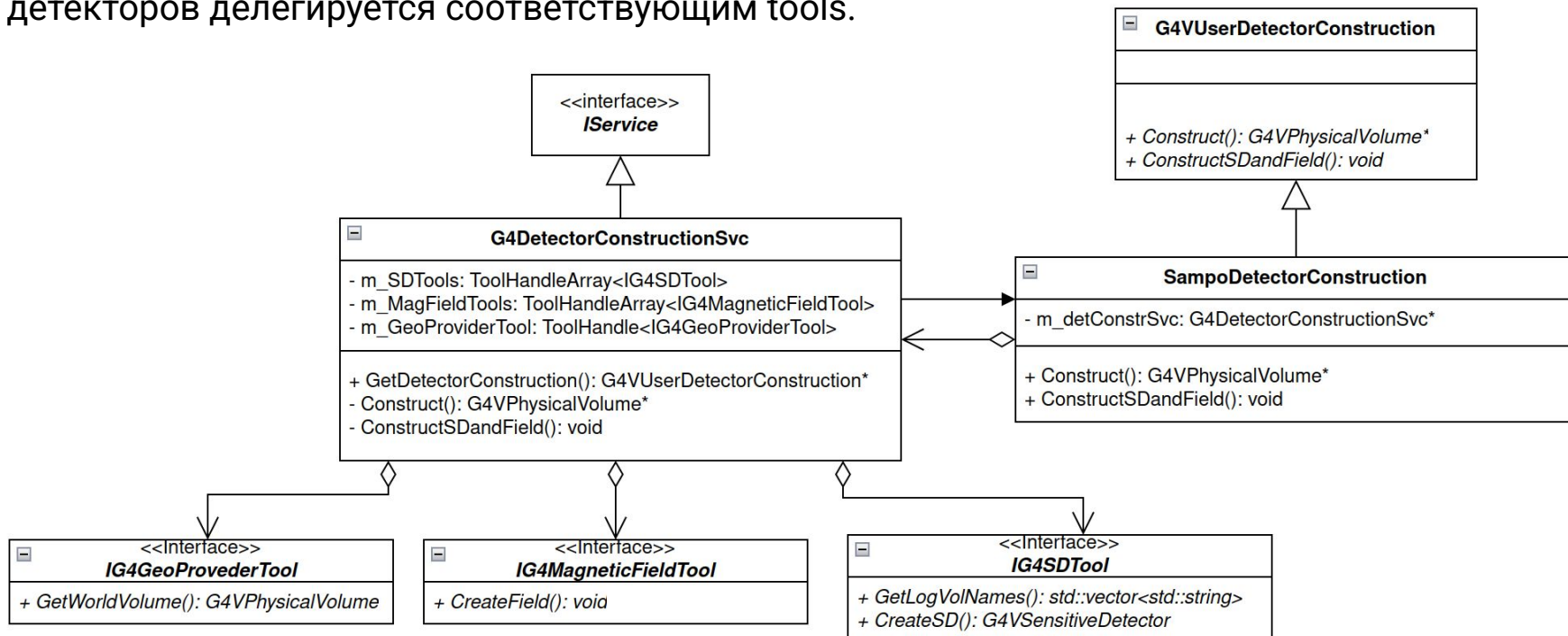
G4PhysListSvc

Концепция: G4PhysListSvc создает G4VUserPhysicsList. IG4PhysListTool создает сам PhysList, затем дополнительная физика создается IG4PhysOptionTools.



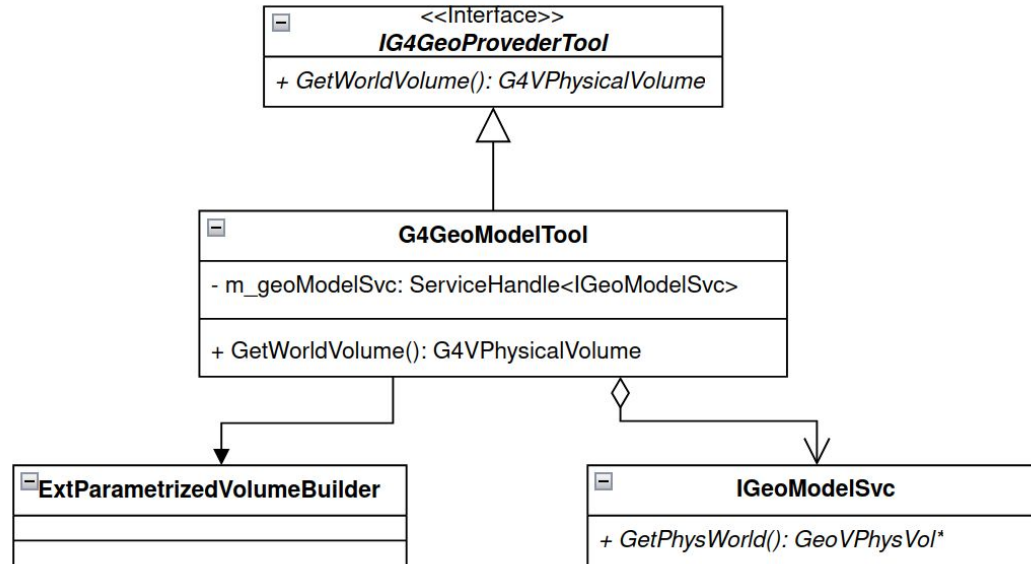
G4DetectorConstructionSvc

Концепция: G4DetectorConstructionSvc оборачивает себя в класс-адаптер SampoDetectorConstruction. Работа по созданию геометрии, поля и чувствительных детекторов делегируется соответствующим tools.

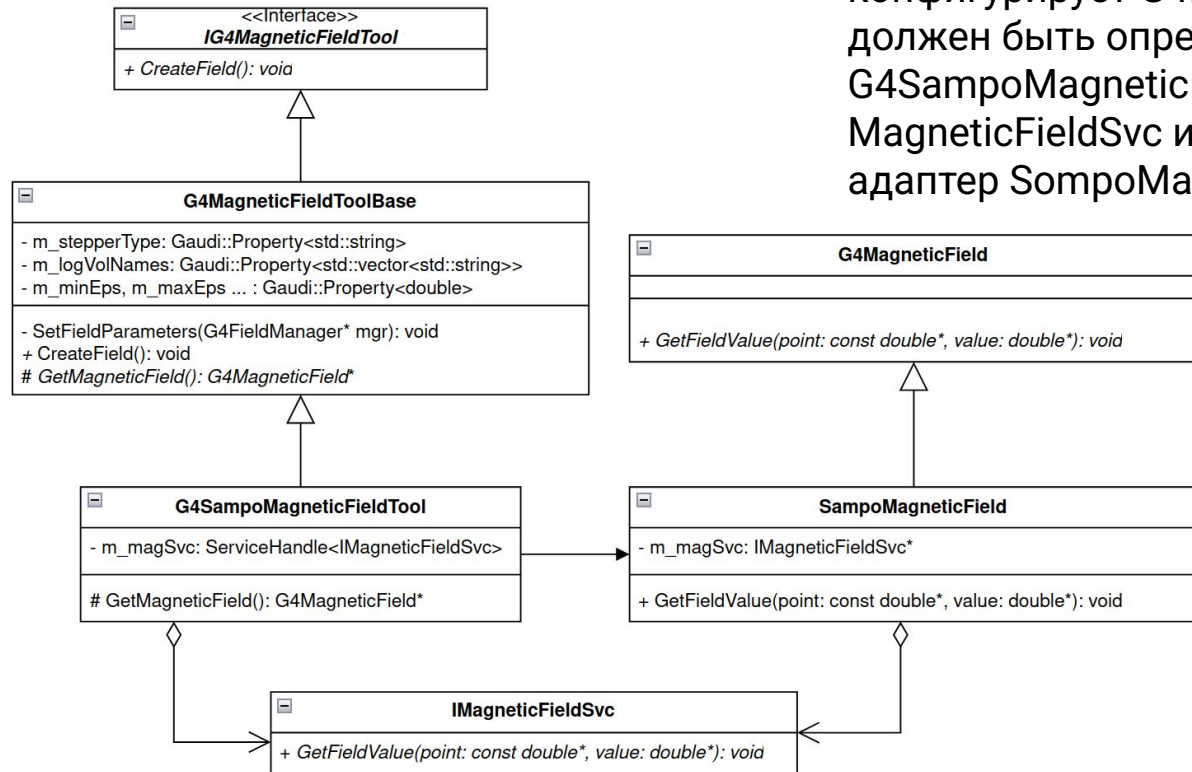


G4GeoModelTool

Концепция: GeoModelTool держит ссылку на GeoModelSvc и умеет конвертировать геометрию из GeoModel в Geant4.



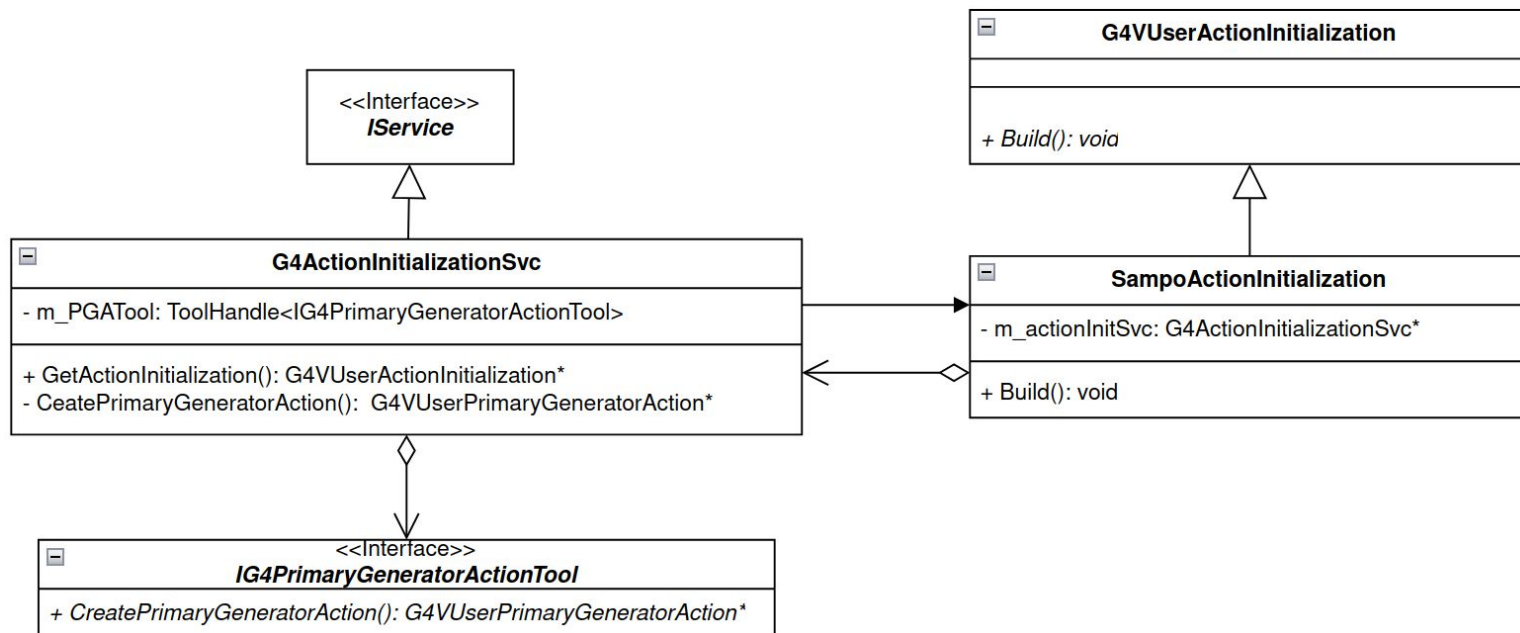
G4SampoMagneticFieldTool



Концепция: G4MagneticFieldToolBase создает и конфигурирует G4FieldManager. *GetMagneticField()* должен быть определен в наследниках. G4SampoMagneticFieldTool держит ссылку на MagneticFieldSvc и оборачивает его в класс-адаптер SampoMagneticField.

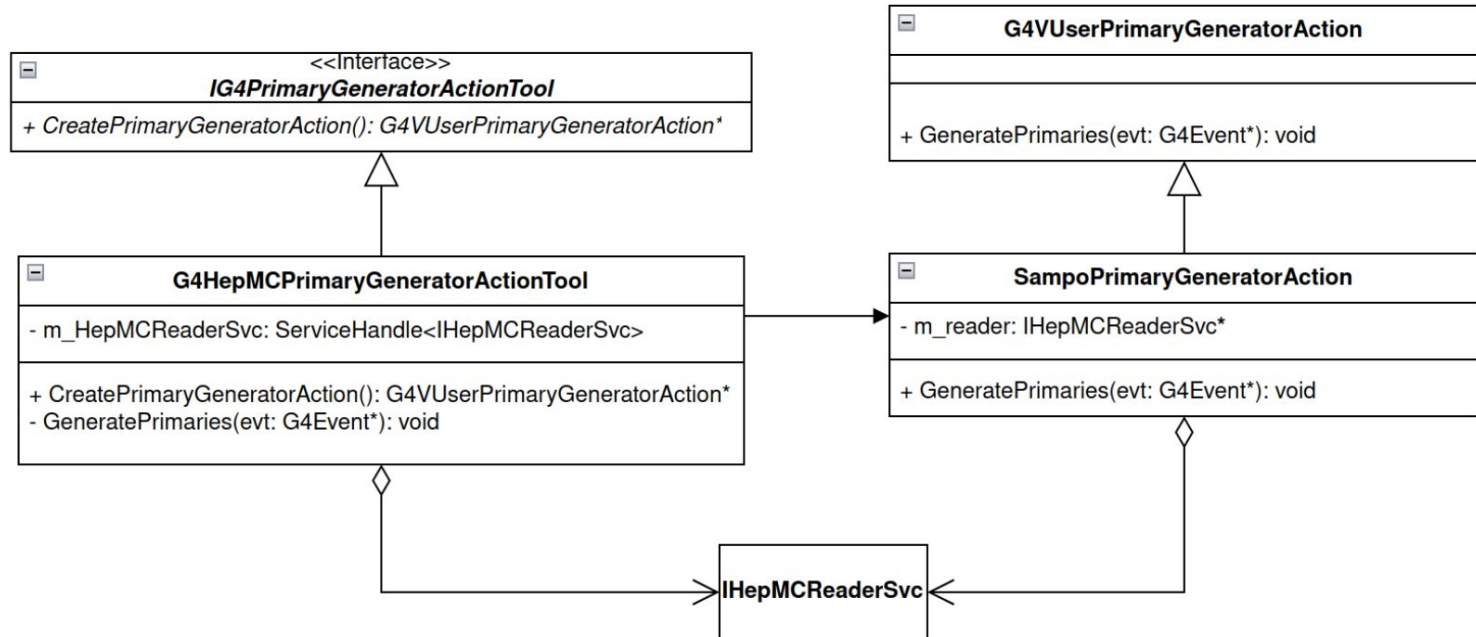
G4ActionInitializationSvc

Концепция: G4ActionInitializationSvc оборачивает себя в класс-адаптер SampoActionInitialization. Работа по созданию хуков делегируется соответствующим tools.



G4HepMCPrimaryGeneratorActionTool

Концепция: G4HepMCPrimaryGeneratorActionTool держит ссылку на HepMCReaderSvc и оборачивает его в класс-адаптер SampoPrimaryGeneratorAction.

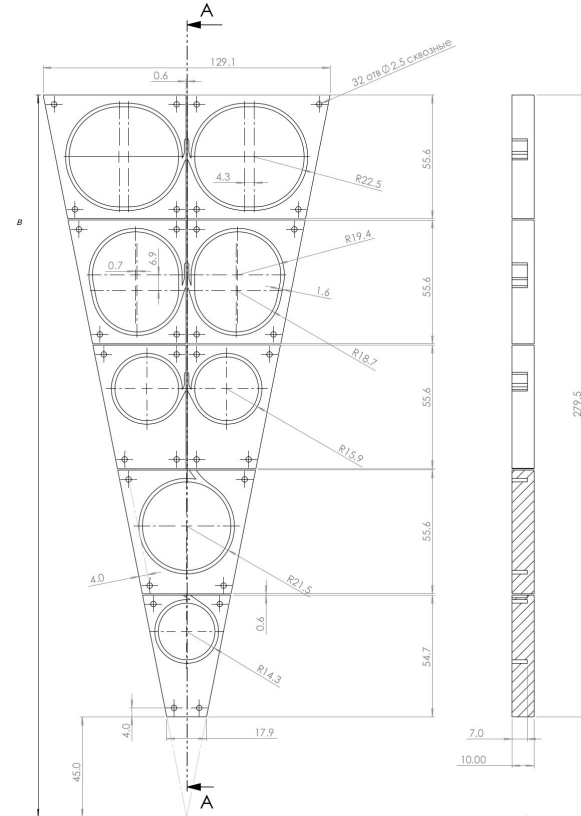
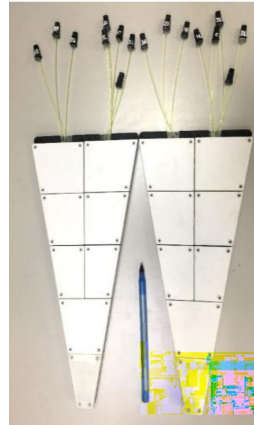
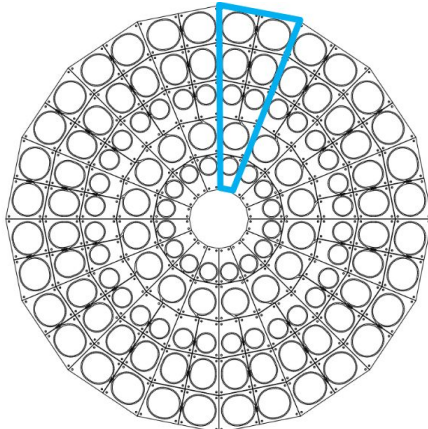


Детектор Beam-Beam Counter (BBC)

Основной задача - измерение азимутальных асимметрий в инклюзивном рождении заряженных частиц для определения поляризации pp пучков.

Задачи в ионной программе - оценка центральности столкновений и определение плоскости реакции.

Фаза 0 - тестирование детектора BBC.

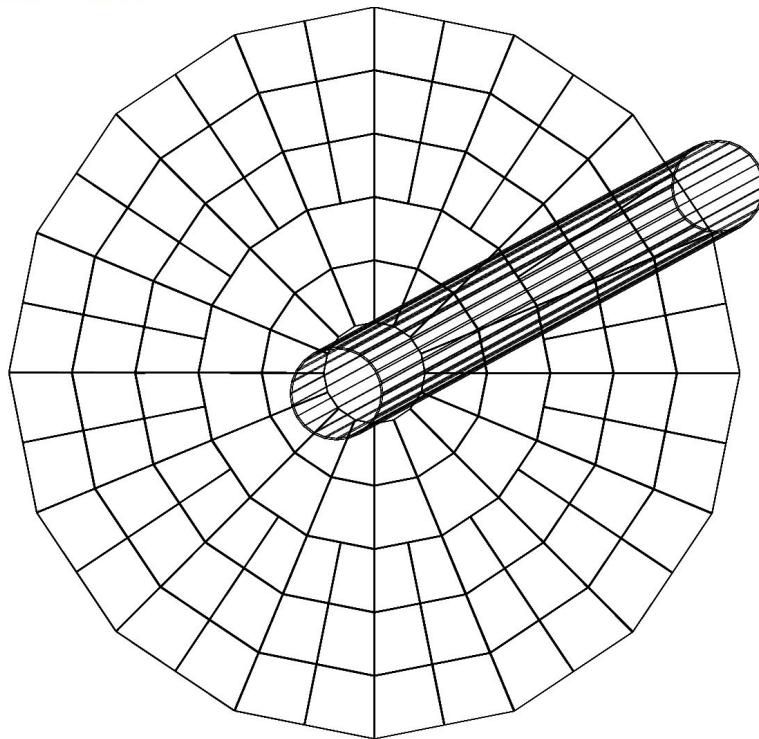


Python JobOptions: моделирование ВВС

```

1 from Configurables import (HepMCReaderSvc, G4SampoAlg, G4DetectorConstructionSvc,
2                             G4PhysListSvc, G4ActionInitializationSvc, BBCGeoProviderTool,
3                             BBCGeoInserterTool, PipeGeoInserterTool,
4                             G4HepMCPrimaryGeneratorActionTool,
5                             BBCPhysListTool, BBCSDTool, GaudiPersistency)
6
7 from Configurables import Gaudi__RootCnvSvc as RootCnvSvc
8 from Gaudi.Configuration import *
9
10 # DetectorConstruction setup
11 detConstrSvc = G4DetectorConstructionSvc()
12
13 # BBC detector
14 bbsGeoInserter = BBCGeoInserterTool()
15 bbsGeoInserter.thickness = 10
16 bbsGeoInserter.distance = 3000
17 bbsGeoInserter.radius_0 = 45
18 # also other BBC parameters here
19
20 # NICA Pipe
21 pipeGeoInserter = PipeGeoInserterTool()
22 pipeGeoInserter.thickness = 1.5
23 pipeGeoInserter.distance = 7500
24 pipeGeoInserter.width = 84
25
26 bbcGeoProvider = BBCGeoProviderTool()
27 bbcGeoProvider.GeoInserters = [bbsGeoInserter, pipeGeoInserter]
28 detConstrSvc.GeoProviderTool = bbcGeoProvider
29

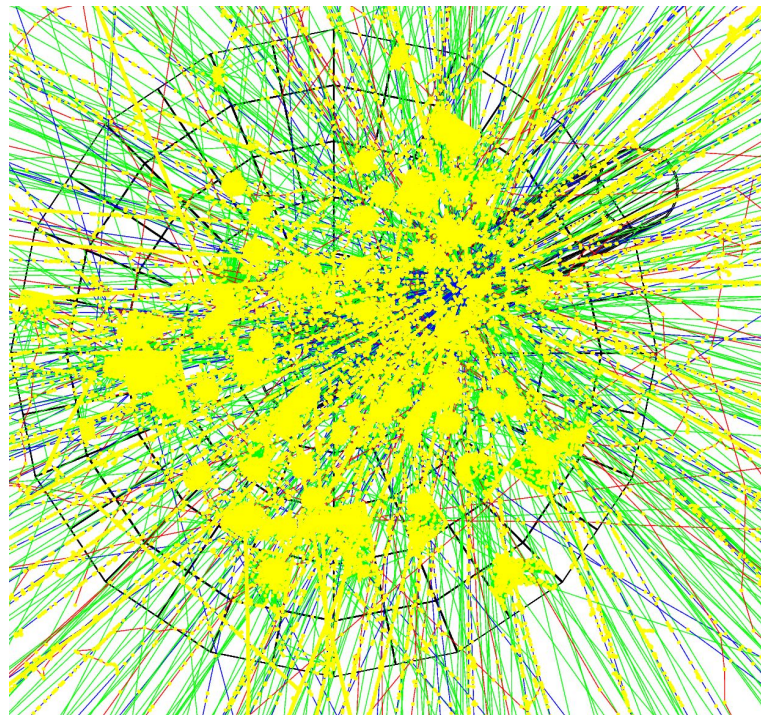
```



Python JobOptions: моделирование BBC

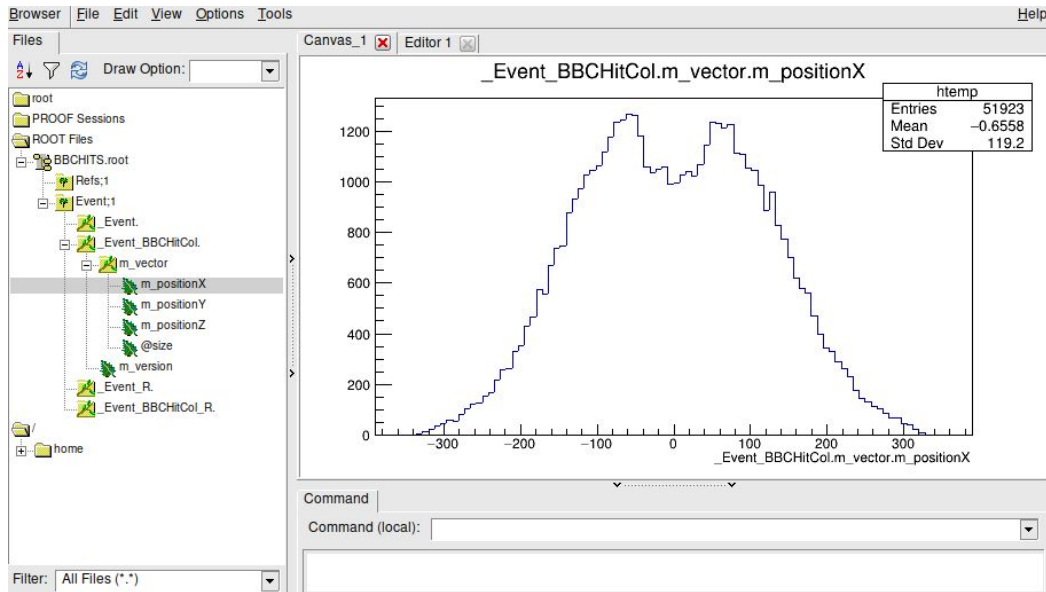
```
30 # PhysList setup
31 physListSvc = G4PhysListSvc()
32 physListSvc.PhysListTool = BBCPhysListTool()
33
34 #HepMC input
35 readerSvc = HepMCReaderSvc()
36 readerSvc.inputFormat = "Ascii"
37 readerSvc.inputFiles = \
38 | ["/workspaces/sampo/BBC/data/urqmd.dat"]
39
40 # ActionInitialization setup
41 ActionInitSvc = G4ActionInitializationSvc()
42 ActionInitSvc.PrimaryGeneratorActionTool = \
43 | G4HepMCPrimaryGeneratorActionTool()
44
```

Столкновения Xe (3 ГэВ/нуклон) с W мишенью
Генератор - URQMD



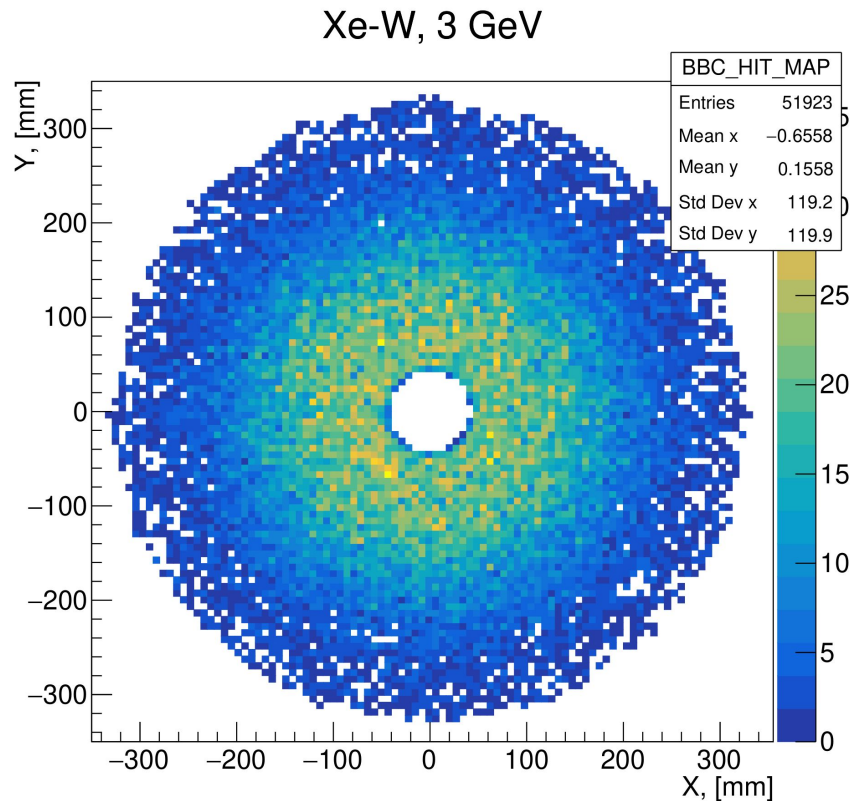
Python JobOptions: моделирование BBC

```
45 # Also add SD to G4DetectorConstructionSvc
46 sdTool = BBCSDTool()
47 sdTool.LogVolNames = ["BBC"]
48 detConstrSvc.SDTools = [sdTool]
49
50 g4algo = G4SampoAlg("G4Alg")
51
52 # Setup writing to ROOT file
53 dst = OutputStream("RootDst")
54 dst.ItemList = ["/Event#999"]
55 dst.Output = ("DATAFILE='PFN:BBCHITS.root' "
56             "SVC='Gaudi::RootCnvSvc' "
57             "OPT='RECREATE'")
58 RootCnvSvc(OutputLevel=INFO)
59 GaudiPersistency()
60
61 evt_max = 1000
62 evt_sel = "NONE"
63
64 # create ApplicationMgr and start Gaudi app (C++)
65 ApplicationMgr(
66     EvtMax=evt_max,
67     EvtSel=evt_sel,
68     TopAlg=[g4algo],
69     OutStream=[dst]
70 )
```



Python JobOptions: моделирование BBC

```
1 from Configurables import Gaudi_RootCnvSvc as RootCnvSvc
2 from Configurables import GaudiPersistency, BBCHitMapAlg
3 from Gaudi.Configuration import *
4
5 esel = EventSelector(PrintFreq=50, FirstEvent=1)
6 esel.Input = [
7     ("DATAFILE='PFN:BBCHITS.root' "
8      "SVC='Gaudi::RootEvtSelector' "
9      "OPT='READ'")
10 ]
11
12 # Output Levels
13 RootCnvSvc(OutputLevel=INFO)
14
15 GaudiPersistency()
16
17 hitMapAlg = BBCHitMapAlg("BBCHitMapAlg")
18
19 # Application setup
20 app = ApplicationMgr()
21
22 # - Algorithms
23 app.TopAlg = [hitMapAlg]
24 app.EvtMax = -1
25 app.HistogramPersistency = "NONE"
```



Заключение:

- Гибкий и масштабируемый дизайн
- Доступна генерация на Pythia8
- HerMC3 основной формат событий
- Описание геометрии через GeoModel
- Включено магнитное поле
- Поддерживается Geant4 (однопоточно)



The Forging of the Sampo, Akseli Gallen-Kallela, 1893