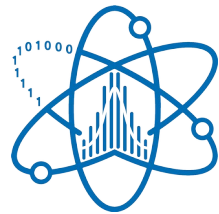




Национальный
исследовательский

**Томский
государственный
университет**



SPDGeoModel development

Makhmanazarov Ramdas

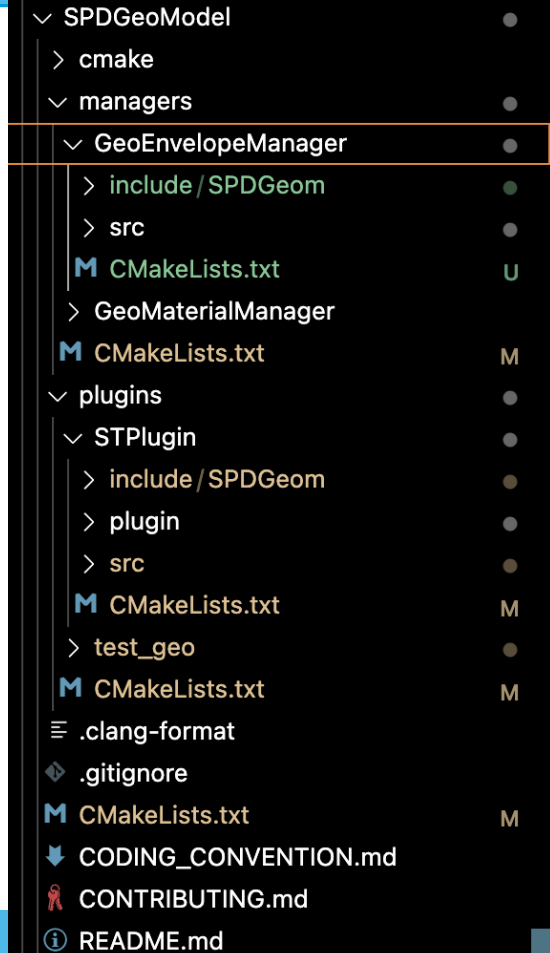
Ramdas Makhmanazarov, on behalf of SPDGeoModel team

Contents

- SPDGeoModel project architecture
 - GeoEnvelopeManager
 - GeoMaterialManager
 - GeoPlugin
 - GeoConstruction (EndCap && Barrel)
- Detector Summary
- Resume

SPDGeoModel architecture

- Dependencies:
 - GeoModelCore, Eigen3, XercesC, nlohmann_json, sqlite, **Geant4**
- Key Components:
 - managers
 - SPDGeom::GeoEnvelopeManager
 - SPDGeom::GeoMaterialManager
 - plugins
 - SPDGeom::STPlugin
 - **... others (13)**
 - CMAKE
 - Clang-format, coding_convention, readme, contributing, CI



SPDGeoModel conventions

1. Структура проекта

Плагин должен иметь четкую структуру:

```
PluginName/  
├── CMakeLists.txt  
├── History  
├── include/  
│   ├── NamespaceName/  
│   │   ├── GeoConstruction.h  
│   │   └── PluginHeader.h  
├── src/  
│   └── GeoConstruction.cpp  
└── plugin/  
    └── PluginEntryPoint.cpp
```

2. Именованние (Naming)

Файлы: `ClassName.h`,
`ClassName.cpp`
(обязательно).

Члены класса: Префикс `m_`
(`m_energy`).

Статические члены:
Префикс `s_`
(рекомендуется).

Пространства имен
(namespaces):
Используются для
избежания конфликтов
(обязательно).

3. Автоматическое форматирование (clang-format)

Проект использует `clang-format`
для единообразия. Ключевые
настройки:

Отступ (Indent): 2 пробела (табы
запрещены).

Column Limit: 100 символов.

Стиль Braces: Перенос фигурных
скобок `{` для классов, функций,
управляющих конструкций.

Пробелы: Логичное и
единообразное использование
вокруг операторов и ключевых
слов.

SPDGeoModel Contributing

1. Непрерывная интеграция (CI)

Сборка и тестирование автоматизированы через GitLab CI.

Используется специальный Docker-образ с настроенным инструментарием (`$ {GEOMODEL_IMAGE}`).

Сборка: Выполняется `make build-release` для Release-конфигурации.

Главные принципы

Ветвление: Вся разработка в ветках, слияние через MR.

Синхронизация: Перед пушем всегда `rebase` на `origin/main`.

Стандартизация: Единая среда сборки (CI) и конфигурации (CMake Presets) для всех разработчиков.

2. Git Workflow: Ветка для каждой задачи

Разработка ведется напрямую в основной репозитории через `feature branches`.

Главная ветка: `main` (в нее напрямую не коммитят!).

Процесс:

```
git checkout -b yourname/feature-name
```

<Do some changes>

```
git add ..., git commit -m "message"
```

Публикация ветки на сервер:

```
git push -u origin yourname/feature-name
```

Создание Merge Request (MR) в GitLab для код-ревью и слияния в `main`.

Envelope Manager

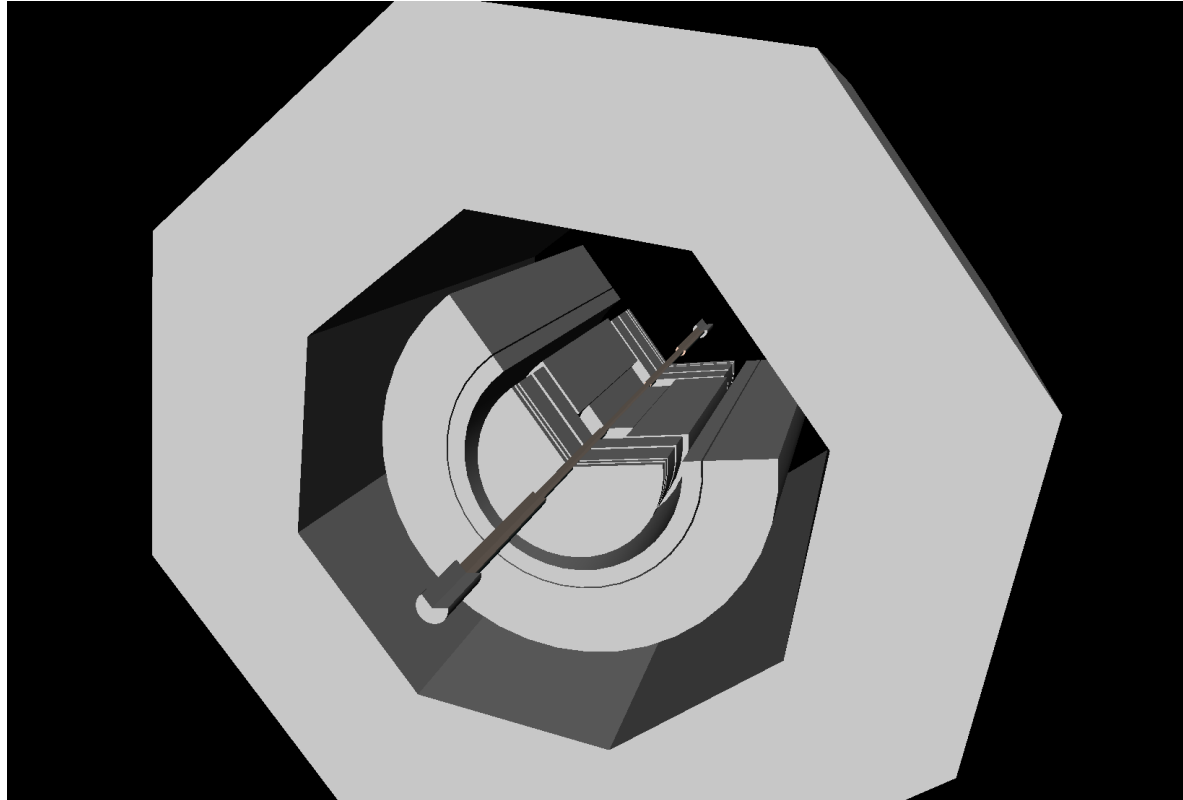
Класс SPDGeom:GeoEnvelopeManager отвечает за:

- Создание материнских объёмов для детекторов.
- Расположение материнских объёмов в мировом объёме
- Предоставление доступа к объёмам через синглтон.

```
// Construct physical volumes for envelopes
//
auto shapeECalBarrel = envmanager->getShape("ECalBarrel");
auto logECalBarrel = new GeoLogVol("/SPD/ECalBarrel", shapeECalBarrel, air);
auto physECalBarrel = new GeoPhysVol(logECalBarrel);

auto shapeECalEndcap = envmanager->getShape("ECalEndcap");
auto logECalEndcap = new GeoLogVol("/SPD/ECalEndcap", shapeECalEndcap, air);
auto physECalEndcap = new GeoPhysVol(logECalEndcap);
```

Envelope Manager



SPD Layout by Evgeny Chernyaev

Material Manager

Класс SPDGeom::GeoMaterialManager будет отвечать за:

- Определение материалов используемых в SPD.
- Предоставление доступа к материалам через синглтон.

```
// Instantiate material and envelope managers
//
auto matmanager = GeoMaterialManager::getManager();
auto envmanager = GeoEnvelopeManager::getManager();

// Get material
//
auto air = matmanager->getMaterial("std::Air");
```

Material Manager

```
1 // clang-format off
2 //=====
3 // m(name, density) - add material
4 // a(name, fraction) - (add) component
5 // c(name, fraction) - concluding component
6 //=====
7
8 //////////////////////////////////////
9 //
10 // Common materials
11 //
12 name_space(std)
13
14 // Simple materials
15 m(Helium,      0.000166322) ; c(He, 1) // z=2
16 m(Beryllium,   1.848 ) ; c(Be, 1) // z=4
17 m(Carbon,      2.    ) ; c(C, 1) // z=6
18 m(Graphite,    2.21  ) ; c(C, 1)
19 m(Aluminium,   2.699 ) ; c(Al, 1) // z=13
20 m(Silicon,     2.33  ) ; c(Si, 1) // z=14
```

```
1 // clang-format off
2 // Generated from G4NistMaterialBuilder
3 //=====
4 // name, symbol, z,  a[g/mole]
5 //=====
6 e(H,      H,      1,    1.00794)
7 e(He,     He,     2,    4.00264)
8 e(Li,     Li,     3,    6.94003)
9 e(Be,     Be,     4,    9.01218)
10 e(B,      B,      5,   10.811 )
11 e(C,      C,      6,   12.0107 )
12 e(N,      N,      7,   14.0068 )
13 e(O,      O,      8,   15.9994 )
14 e(F,      F,      9,   18.9984 )
15 e(Ne,     Ne,    10,   20.18  )
16 e(Na,     Na,    11,   22.9898 )
17 e(Mg,     Mg,    12,   24.305  )
18 e(Al,     Al,    13,   26.9815 )
19 e(Si,     Si,    14,   28.0854 )
20 e(P,      P,     15,   30.9738 )
```

GeoConstruction & Plugin

Классы `.*BarrelGeoConstruction` `.*EndCapGeoConstruction` отвечают за:

- Создание элементов объёмов (физические и логические объемы трубок и прочего)

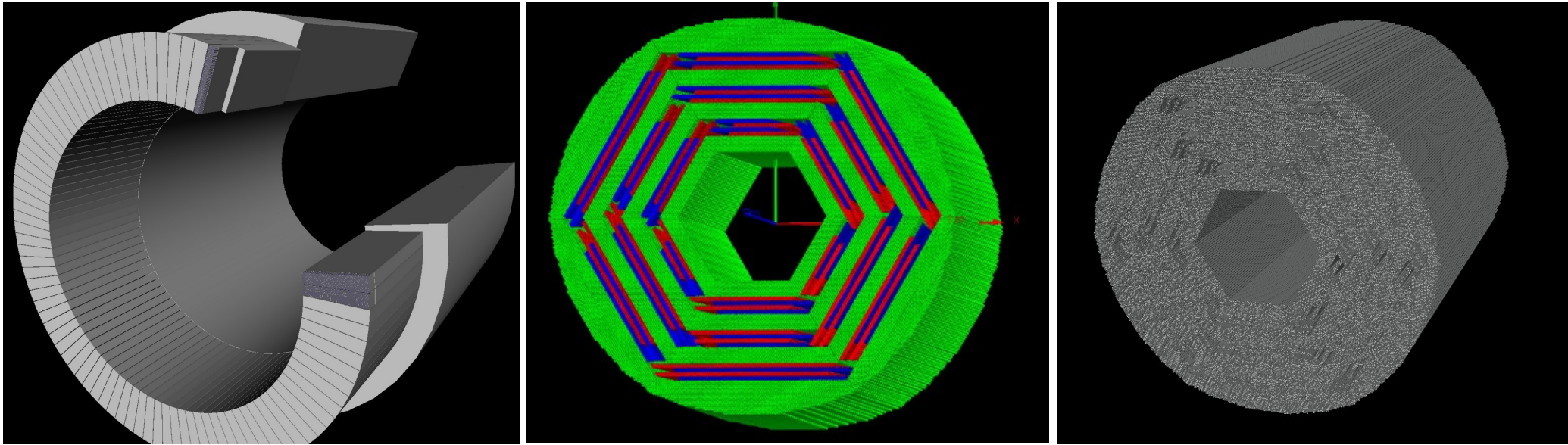
Класс (плагин) `.*Plugin` отвечает за:

- Помещение объектов `Construction` в соответствующий материнский объем
- Создание файла геометрии

```
// Build internal structure of envelopes
//
auto constructorEcalBarrel = new SPDGeom::EcalBarrelGeoConstruction();
constructorEcalBarrel->build(physEcalBarrel);

auto constructorEcalEndcap = new SPDGeom::EcalEndcapGeoConstruction();
constructorEcalEndcap->build(physEcalEndcap);
```

GeoConstruction & Plugin



ECal by Ilia Zimin

Plugins

- 1) SPDGeom::STPlugin (bar, encp)
- 2) SPDGeom::BeamPipePlugin
- 3) SPDGeom::MCPPlugin (cntr, fwd)
- 4) SPDGeom::MCTPlugin (bar)
- 5) SPDGeom::RSPlugin (bar, encp)
- 6) SPDGeom::ZDPlugin (fwd)
- 7) SPDGeom::MSPlugin (bar)
- 8) SPDGeom::VDPlugin (bar, encp)
- 9) SPDGeom::ECalPlugin (bar, encp)
- 10) SPDGeom::TOFPlugin (encp)
- 11) SPDGeom::FARICHPlugin (encp)
- 12) SPDGeom::BBCPlugin (ecnp, fwd)

second stage

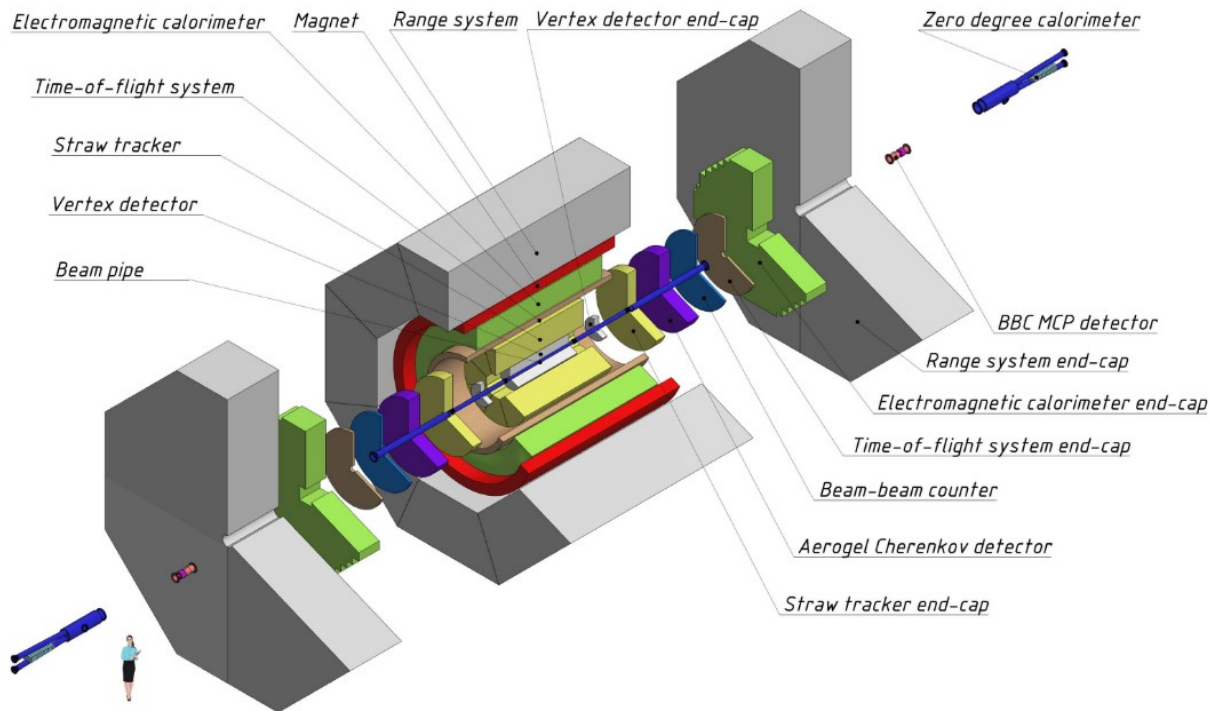
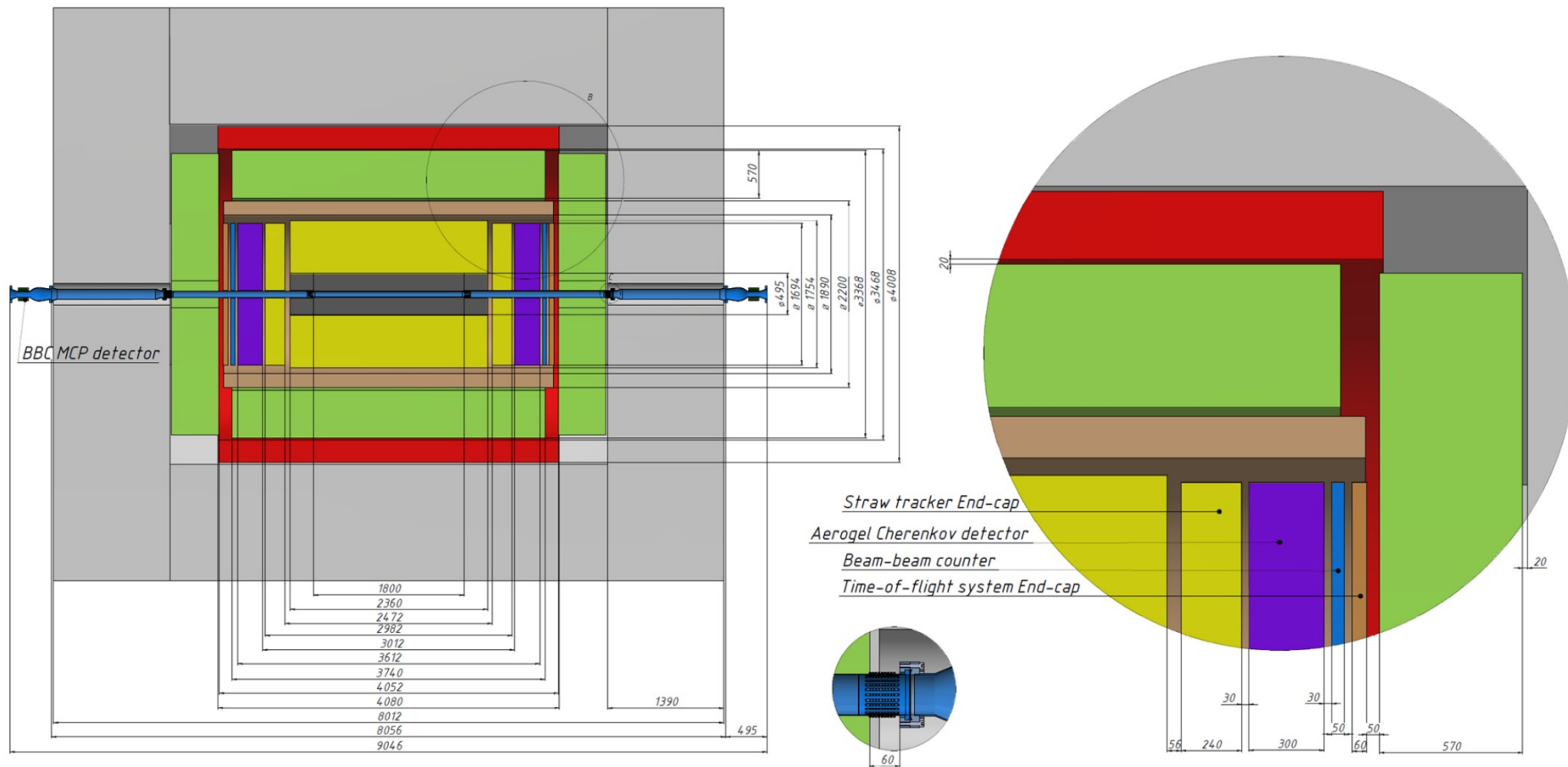


Figure 2.2: Final layout of the SPD setup.

Detector summary



STBarrelGeoConstruction (Octant)

https://indico.jinr.ru/event/5000/contributions/30832/attachments/22228/39241/BarrelSPD_13_05_25.pdf

https://indico.jinr.ru/event/5231/contributions/30361/attachments/21672/38135/Ruslan_SPD&MC_Meeting_2025-02-19.pdf

The mechanical construction proposed for the SPD ST is based on engineering solutions, which were already efficiently applied in the ATLAS and PANDA experiments. The straw tracker consists of three parts: a barrel part and two end-caps. The barrel has inner and outer radii of 270 and 867 mm, respectively. It is subdivided azimuthally into 8 modules, with 31 double layers of straw tubes each, as shown in Fig. 9.19. The layout of a single module with the supporting frame is shown in Fig. 9.20 and Fig. 9.21.

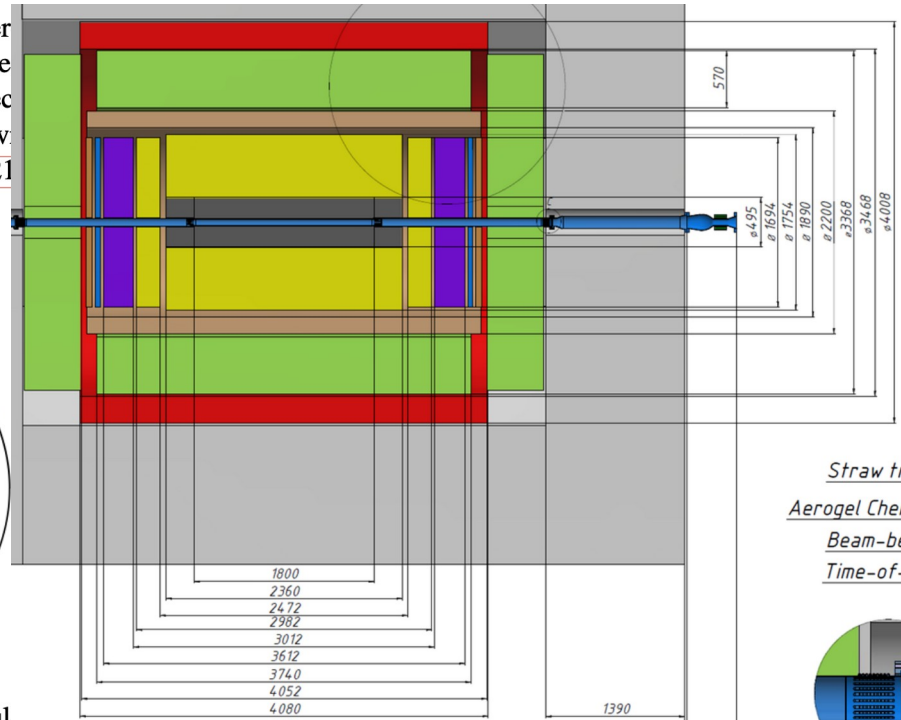
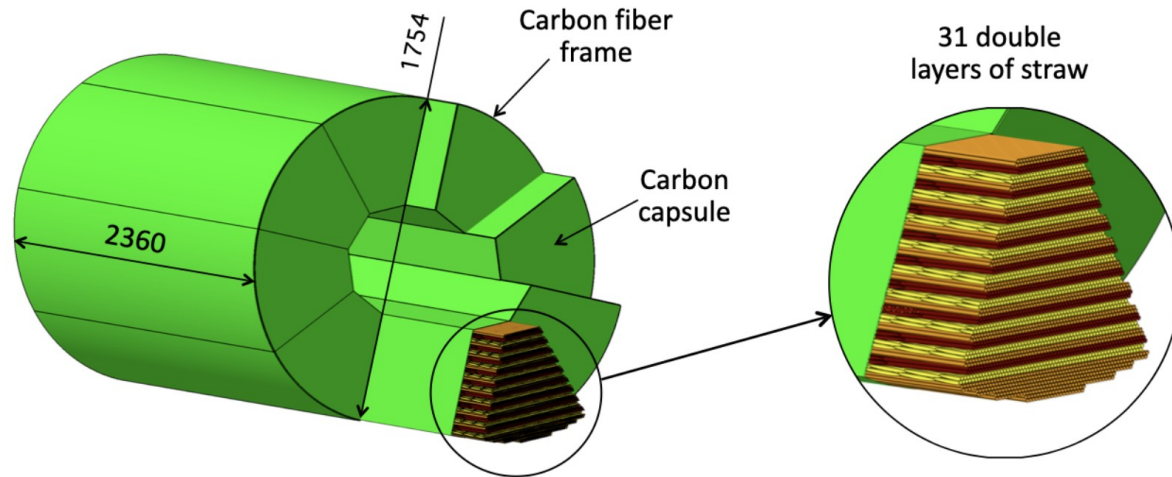


Figure 9.19: General layout of the barrel part of ST, which shows 8 modules (octants). Each module contains 31 double layers of straw tubes encased in a composite-polymer capsule.

Resume

- Зафиксирована и реализована структура проекта SPDGeoModel
- Описаны CMAKE для всех плагинов. Новый плагин можно создать из копии одного из плагинов, поменяв пару строк в CMAKE
- Создано 2 менеджера и ~14 плагинов
- Настроен минимальный *CI Pipeline* в окружении SPDSOFTWARE (Спасибо Льву и Ринату ОИЯИ)
- Готов (на ~90%) плагин STSextantGeoPlugin и GeoConstruction класс.
- Готов (на ~90%) плагин Ecal (Илья Зимин ОИЯИ)
- Сделано >55 MR
- Обнаружены несоответствия размеров в Figure 3.1 p.20 с данными о детекторах.

Дальнейшие планы:

- Дописать ReadMe
- Определиться с сору number
- Заполнять GeoConstruction классы

SPD GeoModel Team

- Evgeny Chernyaev (TSU, GeoModelAtlas, Geant4)
- Ramdas Makhmanazarov (TSU) efemberg11@mail.ru
- Aytadzh Allakhverdieva (JINR)
- Ilia Zimin (JINR)



E.Chenyaev, I.Denisenko (SPD Physics Coord),
R.Makhmanazarov, D.Olyenik (SPD Software Coord)
L.Simbiriatin (SPD Sampo Framework main dev)