

Integration of Tracking Software

FCC hadron detector meeting 21/01/2016

Julia Hrdinka (TU Wien),
Andreas Salzburger (CERN)

Introduction

LHC - software is well tested and used in existing experiments

- Project ongoing to decouple ATLAS Tracking Software into a common tracking toolkit useable for other experiments:

[a-tracking-sw](#)

(see also Andreas talk <https://indico.cern.ch/event/459865/>)

Double build philosophy

- ATLAS Tracking Software build in *Athena* framework (based on *Gaudi*)
- Common tracking toolkit against the *Gaudi* event processing framework
 - enabled by setting build flag `ATS_GAUDI_BUILD`

Introducing - The ATS project

Tracking					
Geometry (TrkDetDescr)	Event Data Model (TrkEvent)	Extrapolation (TrkExtrapolation)	Fitting (TrkFitter)	Calibration, general (TrkTools)	Alignment (TrkAlignment)

a-tracking-sw
Attempt to encapsulate the ATLAS Tracking software from ATLAS
currently restricted access

★ 0 P 1 SSH HTTPS ssh://git@gitlab.cern.ch:7999/ 🔒 📄 + 🔔 GLOBAL ▾

65 COMMITS 5 BRANCHES 0 TAGS 13.21 MB

ADD CHANGELOG ADD LICENSE ADD CONTRIBUTION GUIDE

A Tracking Software (ATS) Project

This library is based on the track reconstruction software developed by the [ATLAS Collaboration](#).

The main philosophy is to provide high-level track reconstruction modules that can be specified for detector technologies by simple extension.

- Event Data Model (EDM)
- Geometry
- track reconstruction tools

The library is attempted to build against **Gaudi** and **Gaudi-Athena**, while additional external dependencies are kept at a minimum.

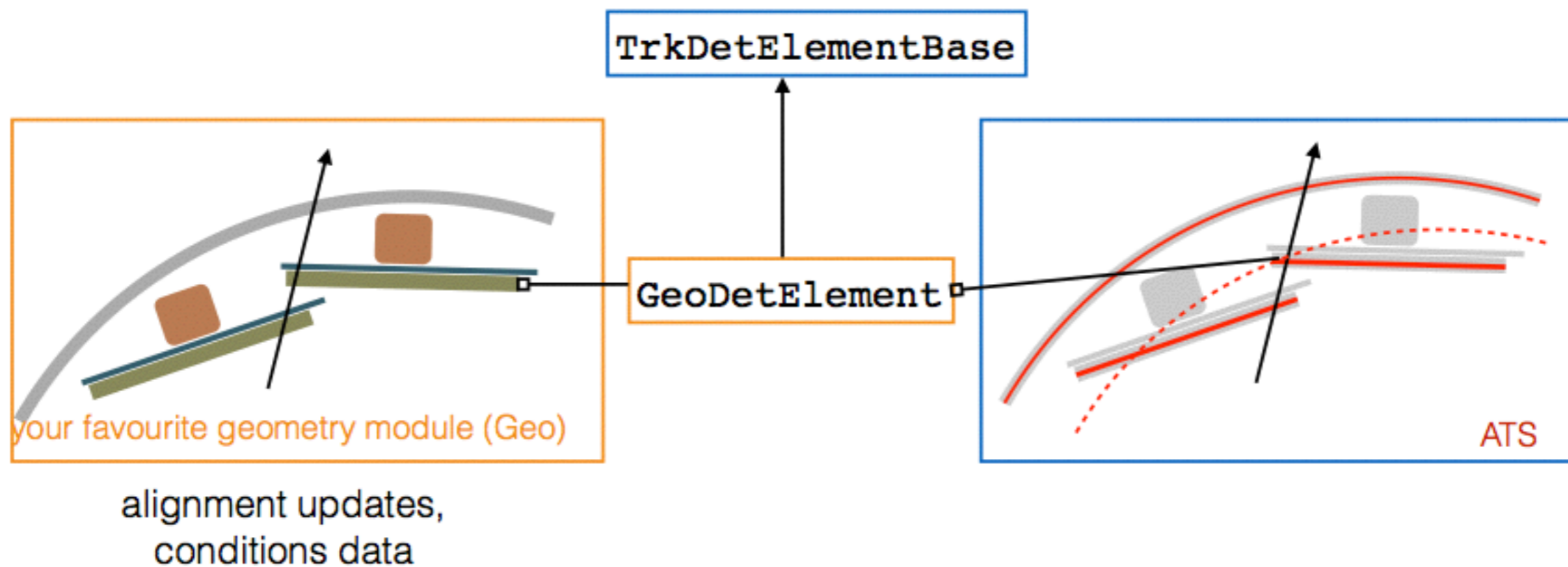
double build philosophy

ATLAS Tracking Geometry - summary

- ▶ ATLAS Tracking Geometry is a geometry model designed around a few basic concepts:
 - core component **Trk::Surface** class (all nodes extend, contain surfaces)
 - embedded navigation through interlinking and volume glueing
- ▶ **Trk::Surface** class builds the core of the geometry and the Tracking Event Data Model (EDM)
 - track parameterisations are done on with respect to a surface
 - extrapolation, track fitting, measurement parameterisations
- ▶ The Tracking Geometry comes with an Extrapolation tool
 - uses the embedded navigation to move through the geometry
 - uses optimised propagation & material effects integration (point-like, dense)

Provide information of Detector element

- Link between detector element of the full and detailed detector description to the TrackingGeometry via `TrkDetElementBase`
- Surface can be build from `TrkDetElementBase` and has direct link to the building geometry



Concrete implementation for FCCSW

- Implement `DD4hepDetElement` inheriting from `TrkDetElementBase`
- The `DD4hepDetElement` serves as an interface to the information provided by DD4hep for the tracking software

```
namespace Det {  
    class DD4hepDetElement : public Trk::TrkDetElementBase {  
    public:  
        /**Constructor*/  
        DD4hepDetElement(const DD4hep::Geometry::DetElement& detElement, std::shared_ptr<const Trk::SurfaceBounds> bounds, const  
            DD4hep::Geometry::Segmentation& segmentation, std::shared_ptr<const Amg::Transform3D> motherTransform = nullptr);  
        /**Destructor*/  
        virtual ~DD4hepDetElement();  
        /** Identifier */  
        virtual Identifier identify() const override;  
  
        /**Identifier hash*/  
        // virtual IdentifierHash identifyHash() const = 0;  
  
        /**Return local to global transform*/  
        virtual const Amg::Transform3D & transform() const override;  
  
        /**Return local to global transform associated with this identifier*/  
        virtual const Amg::Transform3D & transform(const Identifier& id) const override; //need to be changed  
  
        /**Return surface associated with this detector element*/  
        virtual const Trk::Surface & surface () const override;  
    };  
}
```

Geometry Translation

Tracking geometry is accessed over the interface `ITrackingGeometrySvc`

Implementation of a dedicated service which provides the automatically translated geometry

- tools needed which scan through all volumes, layers, surfaces and build them from bottom to top
 - ➔ finally receiving single world tracking volume
- special building tools provided by the tracking package help to simplify the translation, examples:
 - ➔ binning arrays of layers inside volumes
 - ➔ building container volumes
 - ➔ automatic interlinking and glueing of volumes

Usage in FCCSW

a-tracking-sw is an external package build with cmake

dependencies

- Event processing framework: Gaudi
- Math library: Eigen
- Identifier: use from DD4hep

connecting links

- **DD4hepDetElement**: providing the detector information to the tracking software
- Translation Service: provides the tracking geometry and converts the full DD4hep geometry automatically into the tracking geometry (possibly using tools for general translation)

Current Status & Outlook

Tracking Software

- Core classes of packages `TrkDetDescr`, `TrkEvent` and `TrkExtrapolation` build against Gaudi(CMake) and Athena(cmt)
- Further Integration of missing classes
- Integration of Kalmanfitter started
- started building first dummy detector

Current Status & Outlook

Integration in FCCSW

- Begin testing of first iteration of translation service with dummy detector and straight line intersection and displaying the geometry
- For FCC Rome workshop 2016: demonstrator including Geometry, fast simulation, truth tracking and track fitting planned