aidaTT

Status of Tracking Toolkit

Ch. Rosemann

DESY

27. March 2014
AIDA annual meeting

aidaTT in a nutshell

Toolkit

- Track fitting functionality $\approx 80\%$
- Track finding functionality (virtual 50%)

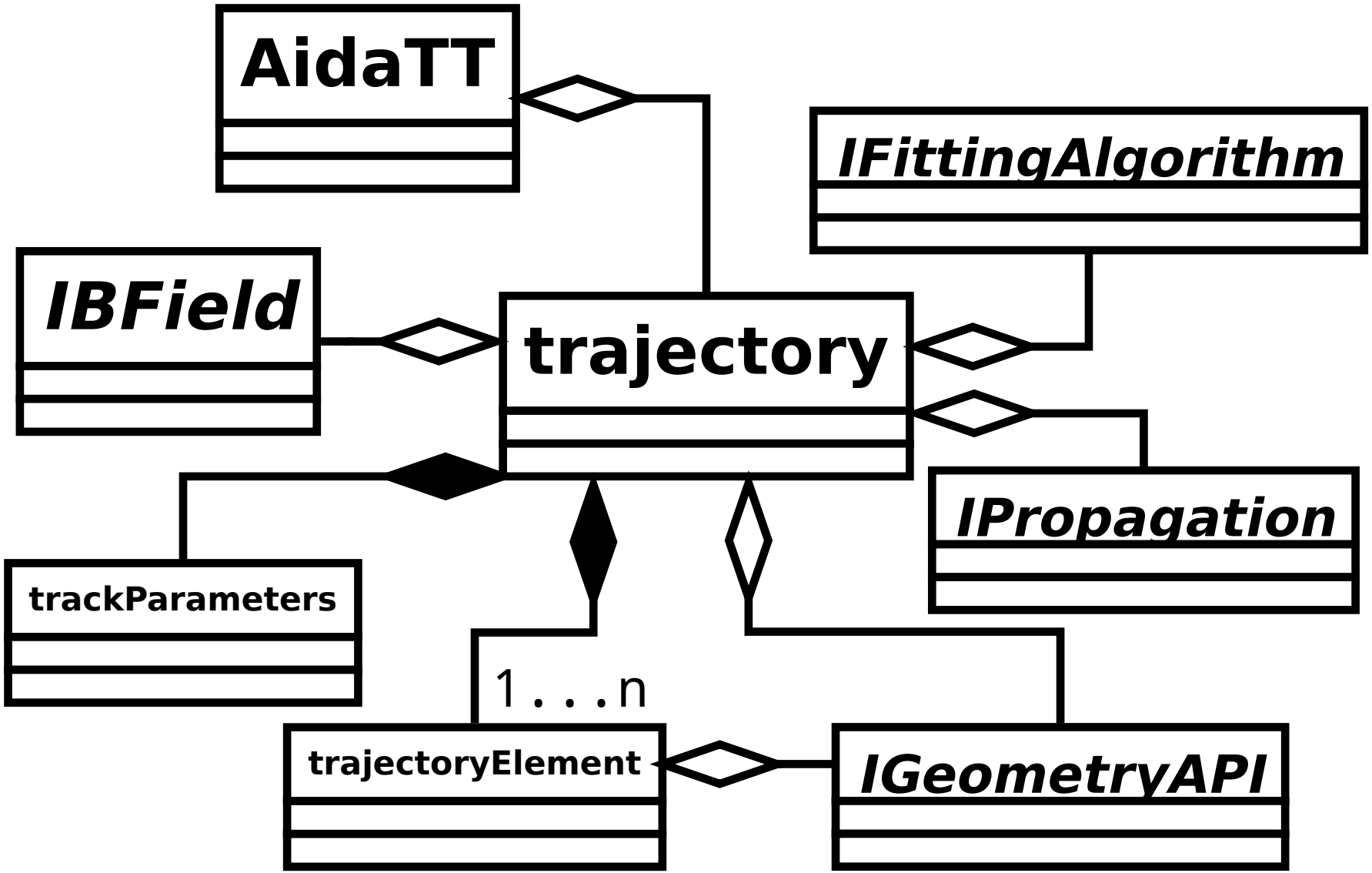
aidaTT extends and succeeds IMarlinTrk

- IMarlinTrk focussed on Kalman Filter
- specifically tailored around one implementation

Computational Design

- Completely modular
- Clear API to reco frameworks
- Complete separation of data, algorithms and functionality
- Parallelization on single track level possible

aidaTT core UML (iteration # 6)



Implementation

IBField

- constant B field

IFittingAlgorithm

- General Broken Lines (about 80% done)

IGeometry

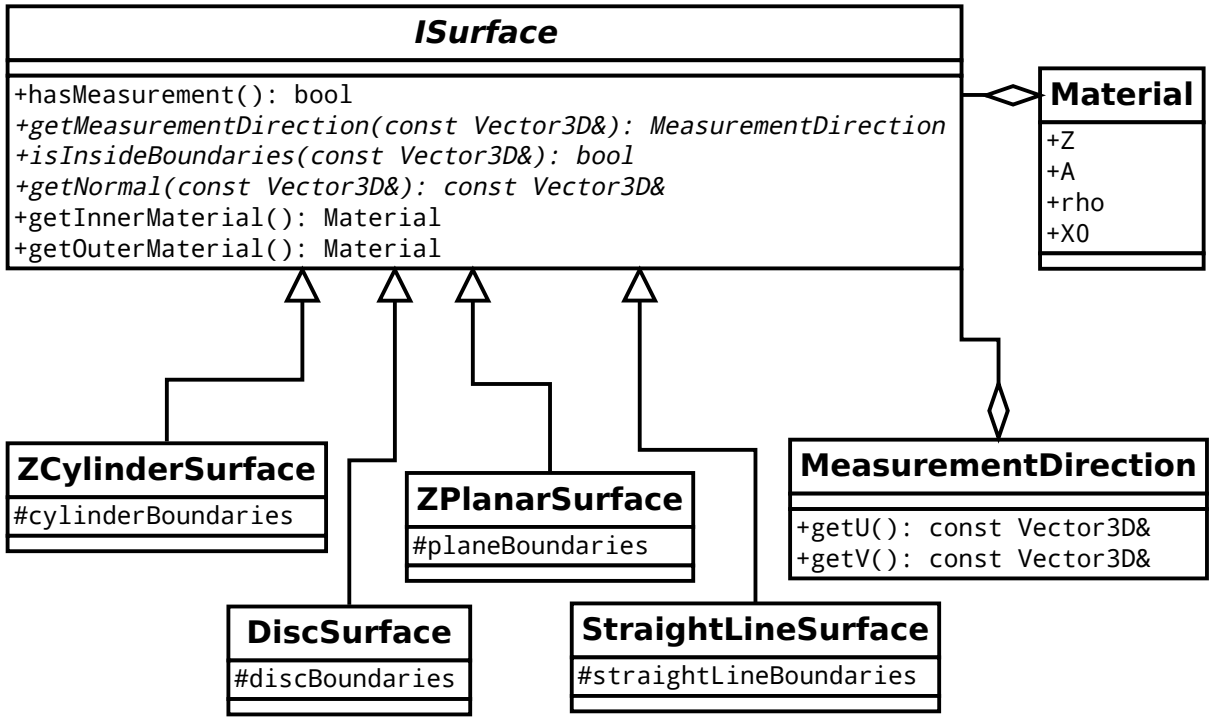
- `dd4hep::DDSurfaces` (although still largely untested)

IPropagation

How to move from point to point on trajectory, fixes track model

- Simplified Propagation (quadratic in arc length)
- Analytical Propagation (perfect helix in homogeneous B field)

dd4hep::ISurface definition



dd4hep::ISurface

- most recent addition, first starting point – maybe volumes later
- geometry provides shapes & boundaries, answer to isInside?, material info, normal vectors and measurement directions
- tracking provides intersection calculation

Central API

`aidaTT::AidaTT`

- master interface to create `aidaTT::trajectory` objects
- instantiates the specific objects: geometry, propagation, fields

`trajectory`

- created/configured with `aidaTT::AidaTT`
- holds a set of track parameters (Kalman lingo: track states)
  23 parameters: $5 \oplus 15 \oplus 3$
- allows fitting of trajectory
- allows adding/removing points/elements from the trajectory
- provides methods for:
  - extrapolation (no material effects)
  - propagation (including material effects)
  - intersecting with **basic** surfaces

Internal implementation

`trackParameters`

- data class to store the 23 parameters
- used in external and internal interface
- additional helper class allows usage of different parametrizations

`trajectoryElement`

- created and controlled by `aidaTT::trajectory`
- placeholder object to anything that belongs to a trajectory
- identified by arc length, wrt reference point of trajectory
- holds Jacobian to next element
- extended by material information (by surface class)
- extended by measurement information (by surface class)
  corresponding surface initialized by `dd4hep::DDSegmentation`

Technical details

Building

- CMake build system
- Different targets: default/install, examples, tests
- Unit test framework included
- svn repository:
  `https://svnsrv.desy.de/public/aidasoft/aidaTT`
- No release yet

dependencies

required:

- gsl – for internal matrix and vector calculation

optional (basic/dummy constructs are still inside):

- GBL – for fitting functionality
- dd4hep – for geometry abstraction

Outlook/Timeline

- Current phase: implementation of full fitting functionality
  still occasional re-iteration of the design
- Documentation available in code (doxygen), several documents on requirements and implementation details, Manual is evolving
- Expectation: initial complete fitting done by end of April
- Open items:
  - needed: finding functionality – maybe June?
  - settling geometry requirements – ongoing process
  - optional: Kalman Filter implementation
  - optional: micro-stepping propagation