AIDA
Advanced European Infrastructures for Detectors at Accelerators

Deliverable Report

SOFTWARE DESIGN FOR TRACKING TOOLKIT

Aplin, Steven (DESY) *et al*

06 June 2012

The research leading to these results has received funding from the European Commission under the FP7 Research Infrastructures project AIDA, grant agreement no. 262025.

This work is part of AIDA Work Package **2: Common software tools**.

Grant Agreement No: 262025

AIDA

Advanced European Infrastructures for Detectors at Accelerators

Seventh Framework Programme, Capacities Specific Programme, Research Infrastructures,
Combination of Collaborative Project and Coordination and Support Action

DELIVERABLE REPORT

SOFTWARE DESIGN FOR TRACKING TOOLKIT

DELIVERABLE: D2.4

Document identifier:	AIDA-Del-D2-4-v2.1
Due date of deliverable:	End of Month 12 (January 2012)
Report release date:	15/05/2012
Work package:	WP2: Common Software Tools
Lead beneficiary:	DESY
Document status:	Draft [Final when fully approved]

Abstract:

A design for a Tracking Toolkit is described. The design foresees the use of a self-contained software framework based on a modern modular design employing well-defined APIs, Event Data Model and geometry description, which is interfaced to the Geometry Toolkit of WP2 task 2.2. The design is currently being prototyped within the tracking software written for the ILD Detector Baseline Document.

Delivery Slip

	Name	**Partner**	**Date**
Authored by	S. J. Aplin and F. Gaede	DESY	15/05/12
Edited by	S. J. Aplin and F. Gaede	DESY	15/05/12
Reviewed by	F. Gaede [WP coordinator] P. Mato [WP coordinator] L. Serin [Scientific coordinator]	DESY CERN CNRS	29/05/12
Approved by	Steering Committee		06/06/12

TABLE OF CONTENTS

Executive summary

The reconstruction of the momentum and path of charged particles forms an essential part of data processing in high energy physics experiments.

The planning of the next generation of high energy physics experiments requires evaluation of the detector performance at a very early stage in the lifecycle of the experiment. This together with the fact that modern tracking algorithms are increasingly complex, requiring very efficient utilisation of computing resources, means that there is a need for a highly flexible Tracking Toolkit which is independent of existing experimental detectors setups and software environments.

The design of such a Tracking Toolkit, which will work in conjunction with the Geometry Toolkit from AIDA WP2 task 2.2, is described in this report. The design uses well-defined APIs, Event Data Model and geometry description, interfaced to the Geometry Toolkit.

The design is currently being prototyped in the tracking software written for the ILD Detector Baseline Document, which will be submitted in 2012. Following this the design will be review such that any issues that arise during this process are properly addressed in the final design.

1. INTRODUCTION

The goal of the task 2.3 within AIDA WP2 is to develop reusable software toolkits that can be adopted in the development of reconstruction software and design studies for future accelerator based particle physics projects. The development of a generic tracking software toolkit forms a key part of this task.

Within modern accelerator based high energy physics experiments, track reconstruction software is responsible for determining the momenta and vertices of charged particles resulting from the primary collisions. The process involves the association of discreet position measurements, so called *hits*, to the trajectory of each individual particle, followed by the subsequent determination of the best estimate of the start, end point, and momenta of the identified particle. The process forms part of the overall event reconstruction and is commonly referred to as *track reconstruction*.

Traditionally track reconstruction has been separated into the two steps identified above, namely pattern recognition algorithms are first employed to sort the hits created by a number of charged particles into a well defined set of track candidates. These are then passed on to a fitting algorithm, which is used to determine the best set of parameters describing each individual track, as well as providing a final validation of the track candidates before they are recorded as particle trajectories for further analysis. Lately there has been some merger of these two steps in modern approaches to track reconstruction, although performance criteria generally mean that the emphasis lies rather clearly with one or the other at any one time during the process.

Currently, while there are generic tracking packages that deal with track fitting [1][2], a solution is lacking that is able to provide both modern pattern recognition and fitting algorithms within a standalone software toolkit/framework. Generally modern tracking software codebases reside in the software frameworks of currently running experiments, where either the question of intellectual property, software dependencies or a lack of available documentation makes their re-use problematic [3]. This, coupled with the complexity of modern track reconstruction algorithms, often makes it difficult for an experiment to perform studies using modern realistic event reconstruction at an early stage in its development cycle.

Such tracking software issues are currently being addressed for the design and optimisation of detectors for the International Linear Collider (ILC) and the Compact Linear Collider (CLIC). These detector concepts are still undergoing both R&D studies as well as overall detector optimisation studies, meaning that the flexibility of the reconstruction tools is of prime importance. A recent complete re-write of the tracking software used in the ILCSoft simulation and reconstruction framework used for these detector studies serves as a prototype for the design of a new tracking toolkit.

The tracking toolkit that will be developed within AIDA WP2 seeks to address the need described above. The design of such a tracking toolkit is outlined below in this report.

2. DESIGN REQUIREMENTS

In order to meet the needs for detector design and optimisation studies the following key criteria will need to be met:

- A well-defined modular approach, based on clearly defined interfaces and APIs;
- Clear separation of framework functionality, data, and algorithmic content, through a well defined Event Data Model and Geometry Description;
- A flexible core framework, which allows efficient definition of different tracking detector setups and geometries;
- Provide well-defined functionality to enable the use of the toolkit with external reconstruction software;
- Easy to add and extend the algorithmic content;
- The utilisation of multi-core architecture needs to be foreseen within the core framework.

3. FRAMEWORK

The software framework will be responsible for the configuration of algorithmic content, managing and providing access to core services, e.g. geometry, and managing the overall flow of data. To meet the needs of the community at which the tracking toolkit is aimed, it will need to have a very limited number of external dependencies to aid easy the adoption of the toolkit within future reconstruction frameworks. This implies that the tracking toolkit will have to provide its own core framework rather than being based on a third party software framework.

Three general components of the framework can be clearly identified:

1. **Algorithms** that run over a set of input data and produce a well-defined set of output data. These algorithms form the foundation of the tracking toolkit.
2. **Services** that are typically only created once and are accessed by any algorithm that requires their functionality, e.g. geometry lookup.
3. **Tools** which are used by the algorithms to perform well defined common actions, which need to be configured individually, even on a per use basis, e.g. a fast and efficient hit sorting function.

The algorithms form the basis of the tracking toolkit. They will perform various well defined actions required during track reconstruction, for instance, hit clustering, track seed formation, or a final fit of the measurements to track parameter. The tracking tool kit will initially contain a set of these algorithms covering all the basic requirements to perform tracking in a generic tracking detector setup.

The framework will be designed such that it is straightforward to create new generic algorithms that have been separated from the method in which the measurements are made, thus allowing these to be added to the toolkit and making them available for re-use. It is foreseen that applicable algorithmic content may already exist in third party software libraries, therefore it essential to provide for efficient interfacing to these existing algorithms, through a strong and clearly defined API.

If multi-threaded programming is to be effectively utilised it will be necessary to foresee this at the very core of the framework. This is especially important regarding the implementation of services where care will be needed to avoid choke points, and inefficient memory utilisation.

Figure 1 illustrates the main components of the Tracking Toolkit. The Algorithms are shown to the left and right of the key concepts of track reconstruction, namely pattern recognition and track fitting, and are built on the Toolkit Framework using the Event Data Model, Geometry Description and core Services.

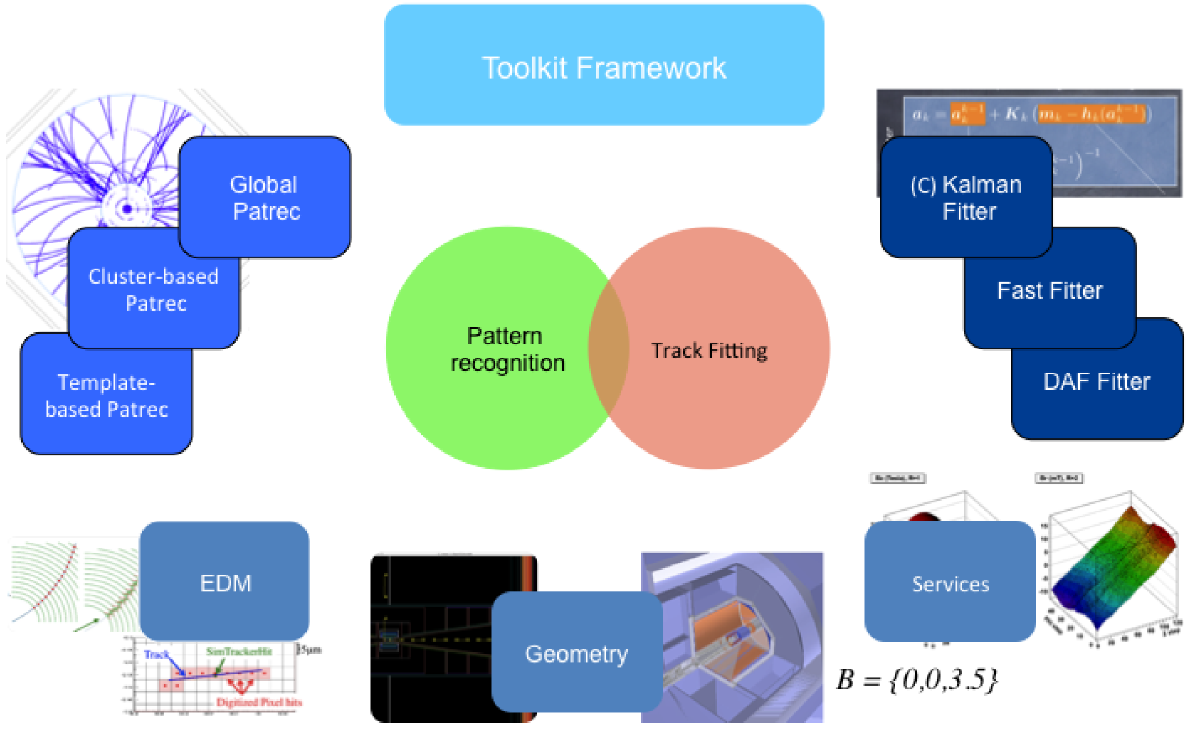

Fig. 1: This figure illustrates the separation of the main components of the Tracking Toolkit design.

4. EVENT DATA MODEL

A common Event Data Model is essential to a modular event reconstruction framework, as it allows for the definition of abstract interfaces by identifying similar tasks to be performed through the method signature and the return type of the method respectively. In addition it provides for various specialised implementations of the same abstract interface base class, thus allowing similar operations to be performed using optimised approaches.

The main components to be defined for a tracking EDM are the position measurements, the hits, and a track class, which represents a collection of those hits, together with the track parameters determined from them. Typically position measurements from tracking detectors are provided within a so-called measurement surface, these may represent a physical surface in the case of a solid-state pixel detector, or a virtual surface defined by geometry e.g. in the case of a Time Projection Chamber. During track fitting such surfaces are required to calculate the residual in local coordinates between the intersection of the surface by a track and the coordinates of a measurement to be included in the fit.

Different track parameterisations would need to be foreseen, with necessary conversion functions available for both the track parameters and their associated covariance. In the case of hits, examples of the usual types would need to be provided, which inherit from a common base class. Whilst it would be sufficient to use only calibrated position measurements for fitting, the needs of pattern recognition may dictate that it is necessary to provide hits at the level of individual pixel, strip or pad readout.

The tracking EDM will use a polymorphic hierarchy of lightweight classes, which enforce a strict separation between data and algorithmic content.

5. GEOMETRY

The requirements of the detector geometry description used in track reconstruction differ from that needed in detailed simulation, for instance such as that used in GEANT4.

Firstly, for performance reasons, the level of detail is generally significantly reduced, with regions of the inactive material within the detector averaged into homogenised volumes, or treated as discrete surfaces at which material effects are accounted for. Secondly, the mapping between position measurements and the individual detector elements needs to be considered, as well as providing for non-physical measurement surfaces as described above. As well as the physical elements of the detector setup, tracking relies heavily on the description magnetic fields. These may be a simplified description of a homogenous solenoid field, using a single parameter to describe the field strength in one dimension, or they may be based on numerical methods employing a field map to describe the combination of a number of highly inhomogeneous fields.

Beyond the geometric description of the detector setup, there is also the access and use of the geometric information to consider. Typical geometric tasks required by tracking algorithms include: global to local coordinate transformations and vice-versa; the calculation of line and surface intersections; as well as the calculation of geometric boundary conditions. Given that a number of modern track fitting and pattern recognition algorithms are iterative means that a large performance increase can be achieved by using an efficient, adaptable and well-tailored geometry system.

Finally, a key concept of tracking algorithms that needs to be addressed is that of navigation, i.e. the efficient traversal of a tree describing the spatial hierarchy of the detectors geometric setup.

It is foreseen that the tracking toolkit will define a high level API for the geometric needs of the tracking algorithms, and that this will be implemented using the Geometry Toolkit from WP2 task 2.2. Figure 1 below shows how a design for such a Detector Description Toolkit may look. In order to use such a design with the tracking toolkit acting as a client RECO Application it would be necessary to create concrete implementation of the RECO Extensions depicted, as well as the Detector constructors if they did not already exist.

The RECO Extensions would need to provide the interface between the Geometry API of the tracking toolkit and the functionality of the Generic Detector Description Model. In addition they would need to provide the aggregation of the detailed level of detector description such that it could be used efficiently for navigation and the treatment of material effects.

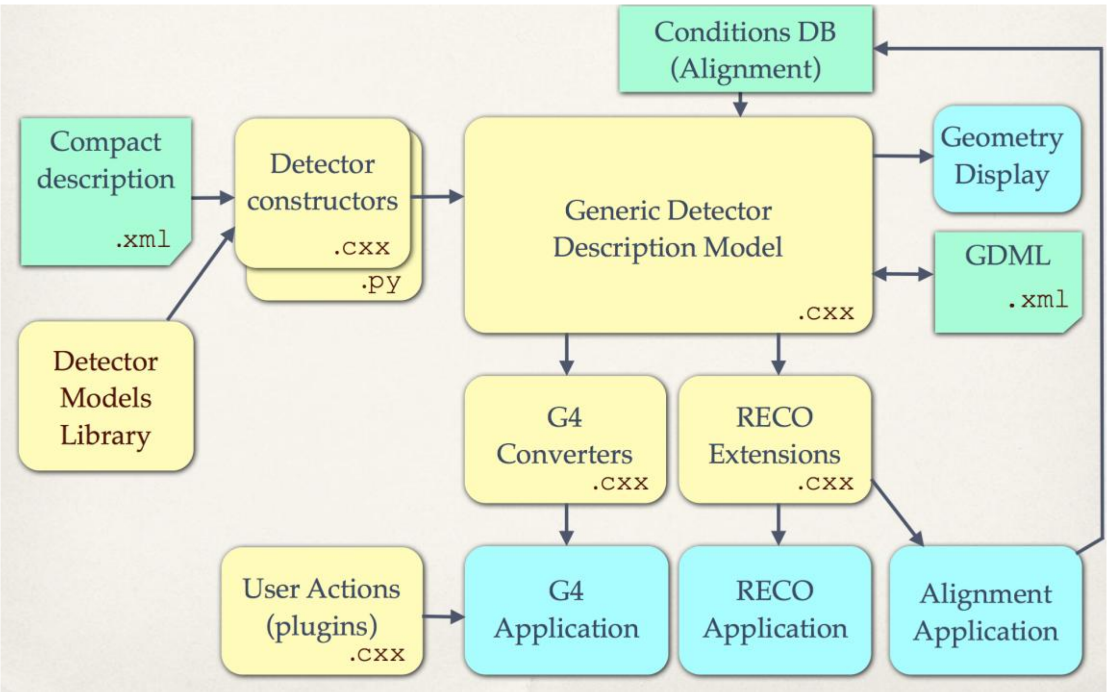

Fig. 2: Design of the detector description toolkit taken from the on-going development in task 2.2.of WP2

6. CONCLUSION

The design of the Tracking Toolkit described above is already being prototyped in the software written to perform the ILD Detector Baseline Document which is due to be submitted at the end of 2012. Once that is finished an evaluation of the current design will be made so that the lessons learned can be fed back into the final design. This will be done in conjunction with the evolution of the design of the Geometry Toolkit in WP1 task 2.2.

7. REFERENCES

JOURNAL ARTICLES:

[1] C. Höppner, S. Neubert, B. Ketzer, S. Paul, A novel generic framework for track fitting in complex detector systems, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 620, Issues 2–3, 11–21 August 2010, Pages 518-525, ISSN 0168-9002, 10.1016/j.nima.2010.03.136. http://www.sciencedirect.com/science/article/pii/S0168900210007473

CONFERENCE PAPERS

[3] Cornelissen, T. et al. (2008) The new ATLAS track reconstruction (NEWT). In: *J. Phys.: Conf. Ser. 119 032014 Proceedings of the International Conference of Computing in High Energy and Nuclear Physics (CHEP07).* Institute of Physics (IOP) Publishing. http://iopscience.iop.org/1742-6596/119/3/032014

WORLD WIDE WEB DOCUMENTS

[2] ACFA Sim-J Group (2012) KalTest: A ROOT-base Kalman Filter Package [online]. Available from: http://www-jlc.kek.jp/subg/offl/kaltest/ [Accessed 15 May 2012].

ANNEX: GLOSSARY

Acronym	**Definition**
ILC	International Linear Collider
CLIC	Compact Linear Collider
EDM	Event Data Model
API	Application Programming Interface