Using DD4Hep through Gaudi for new experiments and LHCb
M. Clemencic and A. Karachaliou on behalf of the LHCb Core Software Team

Abstract
The LHCb Software Framework Gaudi is a C++ software framework for HEP applications used by several experiments. Although Gaudi is extremely flexible and extensible, its adoption is limited by the lack of certain components that are fundamental for the software framework of an experiment, in particular a detector description framework, whose implementation is delegated to the adopters. To enable future experiments to quickly adopt Gaudi, we integrated the DD4Hep toolkit in the existing software framework, and, as a proof of concept, we used it with the LHCb software applications, from simulation to reconstruction and analysis. We will describe how the DD4Hep toolkit can be used by a new experiment, as well as how we can migrate an existing detector description framework to the new toolkit.

Detector Description for HEP (DD4hep) toolkit
- Full Detector Description
  It includes geometry, materials, visualization, readout, alignment, calibration, etc.
- Full Experiment life cycle
  Detector concept development, detector optimization, construction, operation. Easy transition from one phase to the next
- Consistent Description
  Single source of detector information for simulation, reconstruction, analysis, etc.
- Ease of Use
  Few places to enter information. Minimal dependencies

From DD4hep web page http://aidasoft.web.cern.ch/DD4hep

Gaudi
The LHCb Software Framework Gaudi is an experiment-agnostic project providing the necessary interfaces and services for building HEP experiment frameworks in the domain of event data processing applications.

Why DD4hep in Gaudi
Although Gaudi provides the core services required to develop a software framework for an experiment (e.g. messaging, I/O, event processing loop), adopters are required to implement their own Detector Description.

DD4hep in LHCb
We investigated different approaches to adopt DD4hep in LHCb. Unfortunately, the Detector Description framework used in LHCb is based on a memory model orthogonal to the one used in DD4hep. LHCb uses weak references, resolved at runtime, between detectors and volumes, while in DD4hep the connection between detectors and volumes must be established at construction time. This difference practically prevents any possibility of automated migration from the LHCb Detector Description to DD4hep.