



HerdSoftware: a common framework for simulation, reconstruction and analysis

Nicola Mori

INFN Florence



Introduction

- During previous workshop @CERN (Oct 2018):
 - Report of preliminary thoughts on a common SW framework for the collaboration
 - Technical discussion with interested people
 - Coarse definition of requirements and desired features



Introduction

- Design goals:
 - Modern coding approach
 - Shared development
 - Extensibility and configurability
 - Ensure code quality by means of automated tests
 - Exhaustive documentation



Introduction

- Prototyping phase
 - MC simulation with Geant4
 - HERD instrument geometry
 - Reconstruction and analysis
 - Data model, algorithms, I/O
 - Mostly, boring SW engineering stuff...



Introduction

- Development phase
 - MC geometry
 - I/O (Root files)
 - Digitization
 - Tracking
 - ...
- Training phase
 - Hands-on session in Bari (IT) on Oct 2019
 - Representatives from all countries
 - Creation of a data analysis group with bi-weekly meetings
 - Dedicated mailing list

Introduction



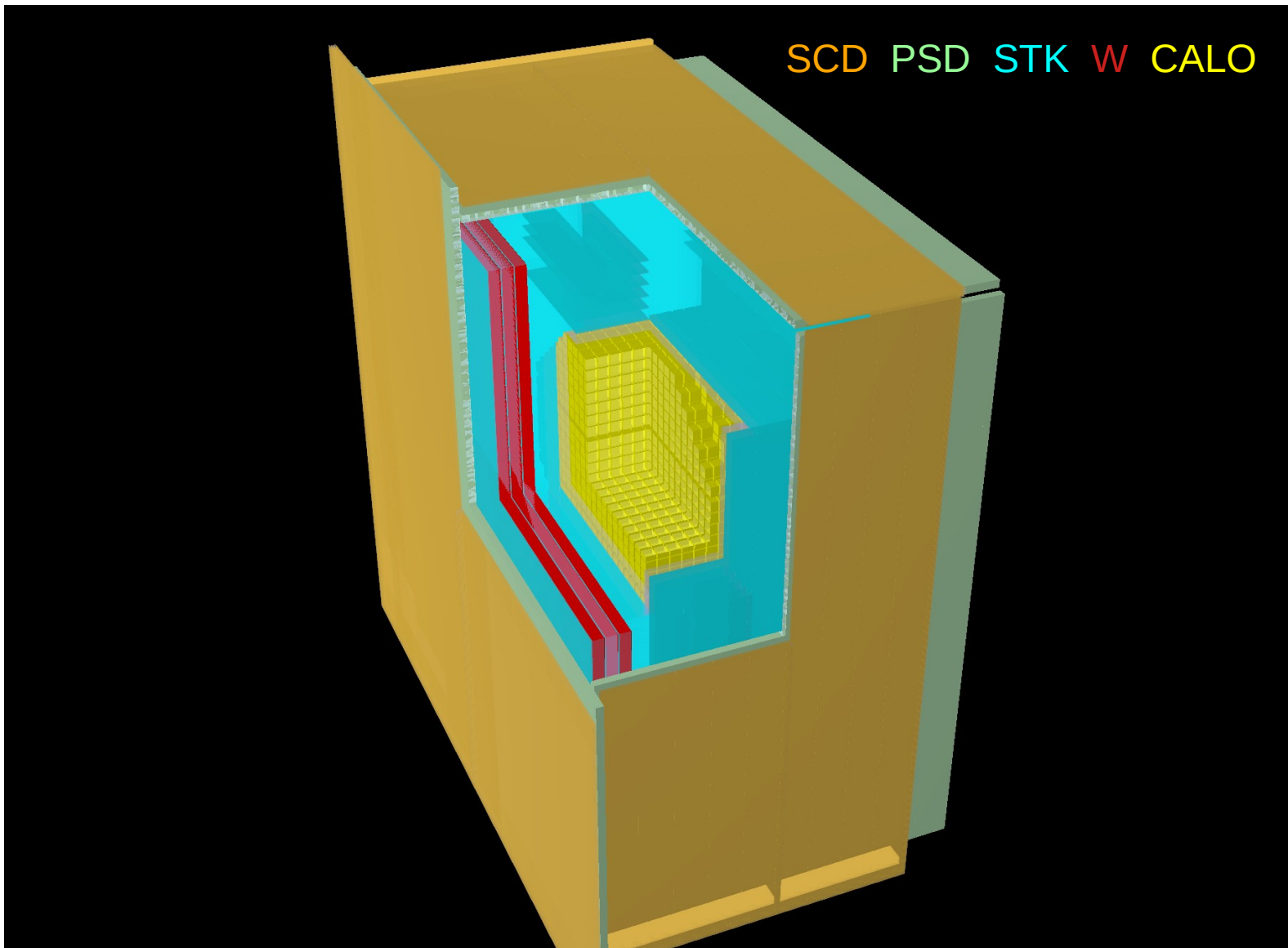
HerdSoftware hands-on session, Bari, 16-18 October 2019



MC simulation

- Based on the GGS framework
 - Executable, particle generation, output, ...
- Detector geometry with configurable parameters
 - Default: CALO+STK+PSD
 - Optional detectors: FIT (replaces STK), SCD

MC simulation





Data processing

- Based on the EventAnalysis framework
 - Generic implementation of the usual HEP data processing procedure
 - Connect to data source (e.g. Root file)
 - Initialize
 - Event loop
 - Finalize
 - Save products of processing (e.g. histograms)



Data processing

- Entities:
 - Data objects (“data model”)
 - Hits, tracks, geometrical parameters, ...
 - Algorithms
 - Digitization, tracking, event display, ...
 - Data provider
 - Read data
 - Persistence
 - Write data
- Encoded as C++ classes



Data processing

- Extensible
 - New algorithms, data objects etc. can be seamlessly added
- Configurable
 - Define a processing pipeline by combining different algorithms
 - Configuration file

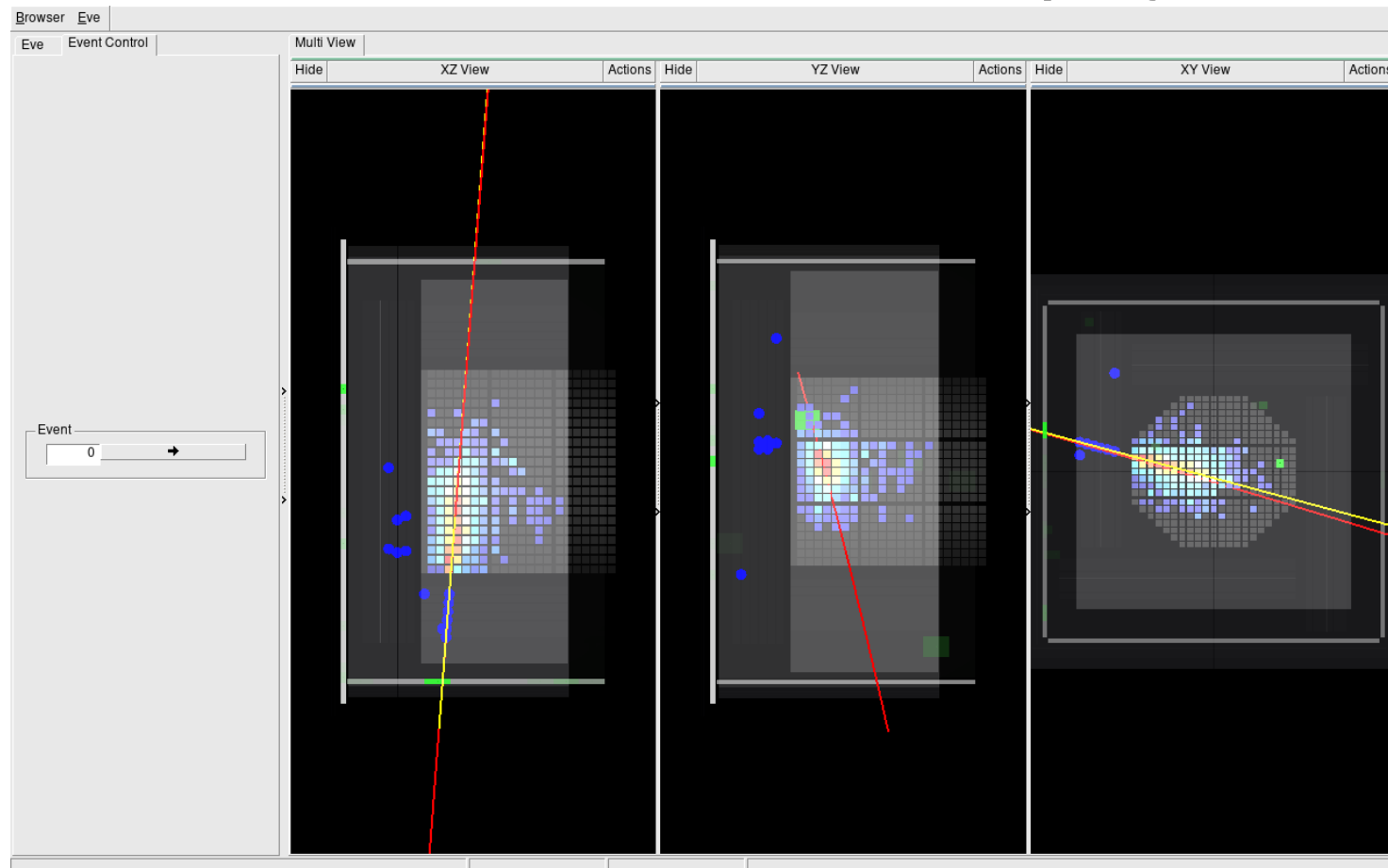


Status

- MC geometry of HERD payload
 - PSD, STK, CALO, SCD, FIT
 - output on Root file
- Data provider for reading MC output files
- Algorithms (computation and selection):
 - Digitization: PSD, STK, SCD
 - Clusterization: CALO, STK, SCD
 - Tracking: Hough transform for STK and SCD
 - Geometry: acceptance, intersections, ...

Status

- Display: HERDward
 - “HERD wants a realistic display”





Status

- Usability
 - Mostly ready for preliminary MC simulations
 - HE ions require some additional effort (integration of HE hadronic engines in Geant4)
 - Physics tuning to be done
 - Mostly ready for reconstruction tasks
 - Algorithms to be improved/added
 - Suitability for analysis tasks is still unclear
 - Reproducible analyses written in a common language, but harder to be coded
 - See L. Pacini's talk for a real-world analysis example
 - Traditional workflow (i.e. Root scripts) is also supported



Infrastructure

- GitLab instance hosted by ReCaS computing center in Bari (Italy)
 - Git repository
 - Issue tracker
 - Continuous integration
 - Wiki with documentation
- Slack channel

Infrastructure

The screenshot displays the main page of the HerdSoftware project on the ReCAS platform. The interface includes a top navigation bar with the ReCAS logo, project navigation options, and a search bar. A left sidebar provides navigation for the project overview, details, activity, releases, cycle analytics, repository, issues, merge requests, CI/CD, operations, packages, wiki, snippets, and settings. The main content area shows the project name 'HerdSoftware' with its ID, statistics (613 commits, 6 branches, 0 tags, 3.7 MB files), and a pipeline status (passed) and coverage (87.40%). A merge request for branch '107-add-clustering-algo-for-scd' is highlighted, showing it was authored by Lorenzo Pacini 1 day ago. Below the merge request are buttons for adding CI/CD configuration, README, LICENSE, CHANGELOG, CONTRIBUTING, and a Kubernetes cluster. A table lists the project's files and folders, including .autoformat, doc, dustbin, examples, include, src, testsuite, and .clang-format, with their last commit messages and update dates.

herd > HerdSoftware > Details

HerdSoftware Project ID: 9

🔔 Star 0 🍴 Fork 2 [Clone](#)

🔍 613 Commits 🌿 6 Branches 🏷️ 0 Tags 📁 3.7 MB Files

pipeline passed coverage 87.40%

master HerdSoftware / +

History 🔍 Find file Web IDE 📄

Merge branch '107-add-clustering-algo-for-scd' cab27d64

Lorenzo Pacini authored 1 day ago

[CI/CD configuration](#) [Add README](#) [Add LICENSE](#) [Add CHANGELOG](#) [Add CONTRIBUTING](#)

[Add Kubernetes cluster](#)

Name	Last commit	Last update
📁 .autoformat	autoformat: return different values for fatal ...	1 month ago
📁 doc	doc: do not generate doxygen documentatio...	10 months ago
📁 dustbin	dustbin: add README.md and MCPPrimaryPa...	2 months ago
📁 examples	algorithms/tracking: fix some bugs, also fix d...	6 days ago
📁 include	Merge branch '107-add-clustering-algo-for-s...	1 day ago
📁 src	Merge branch '107-add-clustering-algo-for-s...	1 day ago
📁 testsuite	algorithms/tracking: fix some bugs, also fix d...	6 days ago
📄 .clang-format	Add code formatting style for clang-format.	1 year ago

Infrastructure

RECAS Projects Groups More

Search or jump to...

HerdSoftware

Project overview

Repository

Issues 26

List

Boards

Labels

Milestones

Merge Requests 0

CI / CD

Operations

Packages

Wiki

Snippets

Settings

Collapse sidebar

herd > HerdSoftware > Issues

Open 26 Closed 87 All 113

Recent searches Search or filter results...

Document how to create dictionaries for data objects #46 · opened 6 months ago by Nicola Mori Documentation To Do updated 6 months ago

Add a "bonding" digitizer algorithm for STK #47 · opened 6 months ago by Nicola Mori Algorithms To Do updated 6 months ago

Check the consistency between different geoParams and Hits 0 of 3 tasks completed #90 · opened 2 months ago by Lorenzo Pacini Rework hits and geometry information Data objects To Do updated 2 months ago

Implement an algorithm for offline trigger #56 · opened 5 months ago by Nicola Mori Algorithms To Do updated 5 months ago

Add a HerdAlgorithm base class for common algorithm features #50 · opened 6 months ago by Nicola Mori Algorithms To Do updated 6 months ago

Double reading of first event #60 · opened 4 months ago by Lorenzo Pacini Bug Providers To Do updated 4 months ago

Implement the FIT in the parametric geometry. #96 · opened 1 month ago by Nicola Mori Monte Carlo To Do updated 1 month ago

Describe the software update procedure in the wiki #86 · opened 2 months ago by Nicola Mori Documentation To Do updated 2 months ago

Add SCD region for delta rays threshold tuning #113 · opened 1 hour ago by Valerio Formato updated 1 hour ago

Check PsdGeoParams test #87 · opened 2 months ago by Lorenzo Pacini Tests To Do updated 2 months ago

Set the segmentation direction in StkGeoParams with StkGeometricDigitizer. 0 of 3 tasks completed #53 · opened 5 months ago by Nicola Mori Algorithms To Do updated 5 months ago

Issues: a list of features to implement and problems to fix

Infrastructure

The screenshot displays a web interface for managing CI/CD pipelines. The top navigation bar includes 'ReCAS', 'Projects', 'Groups', and 'More'. A search bar is present on the right. The left sidebar shows a navigation menu with 'HerdSoftware' selected, and sub-items like 'Project overview', 'Repository', 'Issues', 'Merge Requests', 'CI / CD', 'Pipelines', 'Jobs', 'Schedules', 'Charts', 'Operations', 'Packages', 'Wiki', 'Snippets', and 'Settings'. The main content area shows a list of pipeline runs for the 'herd' project under 'HerdSoftware > Pipelines'. The list includes columns for Status, Pipeline ID, Triggerer, Commit, Stages, and a timestamp. The status of each run is indicated by a colored box: green for 'passed' and red for 'failed'. The 'latest' tag is shown for the most recent run of each pipeline. The list shows several successful runs and several failed runs, with the failed ones having a 'failed' status and a 'retry' icon.

Status	Pipeline	Triggerer	Commit	Stages	Timestamp	Actions
passed	#559 latest		Pmaster -> cab27d64 Merge branch '107-add-cluste...		00:09:53 1 day ago	
passed	#558 latest		P110-fix-tra... -> 1b9bb5a4 analysis/dataobjects: add STKi...		00:09:42 1 day ago	
passed	#557		Pmaster -> 627fea20 Merge branch '111-parametriz...		00:09:16 1 day ago	
passed	#556		P111-paramet... -> e4d6181c geometry/parametricgeo: Fix ...		00:08:55 2 days ago	
failed	#552 latest		PFIT_geometry -> 034601c6 First full version of FIT (top + si...		00:07:05 4 days ago	
failed	#551		PFIT_geometry -> dccbd5cd Top FIT - first clean version of t...		00:06:39 4 days ago	
failed	#550		PFIT_geometry -> 623de51c TOP fit - first version		00:07:06 4 days ago	
passed	#549		P111-paramet... -> 02b2446c geometry/parametricgeo: Add...		00:09:01 5 days ago	
failed	#548		PFIT_geometry -> 6c1c6998 top fit implementation - work i...		00:06:38 5 days ago	
passed	#547		P107-add-clu... -> f429cf67		00:17:39	

Continuous integration: automatically build code and test it on each code commit

Infrastructure

The screenshot shows a web browser displaying a Wiki page for HerdSoftware. The page title is "Download, configure, build and install". The breadcrumb trail is "herd > HerdSoftware > Wiki > Download, configure, build and install". The page is last edited by Nicola Mori 1 month ago. The main content area lists several sections: "Supported platforms", "Build prerequisites" (with sub-sections for Packages, Standard packages, Custom packages, and GGS without Geant4), "Download the code", "Build the code" (with sub-sections for Out-of-source build and Modular build), "Install the package", and "Set the environment". Below this, a paragraph states: "This page contains the instructions about how to obtain the HerdSoftware code, configure and build it." The "Supported platforms" section lists: CentOS 7, Ubuntu 18.04, Ubuntu 16.04, Archlinux, Mac OSX 10.13 "High Sierra", and Mac OSX 10.15 "Catalina". A note mentions that other Linux distributions might work with adjustments. The "Build prerequisites" section states the software is written in C++ using the C++14 dialect and requires a compiler that supports this dialect. The left sidebar shows navigation options like Project overview, Repository, Issues (26), Merge Requests (0), CI / CD, Operations, Packages, Wiki (selected), Snippets, and Settings. The right sidebar shows a "Clone repository" button and a "User's manual" section with a table of contents including: Table of contents, Introduction, Download, configure, build and install, Analysis (with sub-items: Data model, Data providers, Algorithms, Run an analysis, Develop new analysis elements), Simulation (with sub-items: Detector geometry: parametric, Particle generator: isosphere, Run a simulation, Acceptance check in MC), and Examples (with sub-items: Overview, Ex00: produce MC, Ex01: digitize MC, Ex02: analyze MC, Ex03: STK native strips). Below the manual is a "Developer's manual" section with "How to" and "Useful links" sub-sections, and a "More Pages" button.



Summary

- The HerdSoftware framework provides a common SW basis for the HERD collaboration
- It currently allows for:
 - Simulating HERD with different detector configurations
 - Run simple reconstruction tasks
 - Perform data analysis tasks (to some extent)
- A hands-on training session has been held recently
- More features can be coherently implemented, ideas and contributions are welcome/needed

Summary

- Dev team:
 - A few INFN people
 - EU colleagues are starting to join the effort (e.g. FIT code from Geneva)
 - Open to everyone

