DevDet WP2 (Software)

Editor Nigel Hessey

Task 1: Geometry Package
Task 2: Reconstruction Toolkit
Task 3: Simulation
Task 4: Software for Multi-core CPU's

Partners: UK, DESY, CERN, INFN, IN2P3/CNRS, IFIC. (Not confirmed).

WP.2

Particle detectors continuously evolve: future detectors will be bigger with higher granularity, will have higher particle rates, and include new detector types. Also technological advances - especially multicore CPUs - make new demands on the software. These developments require new software features and better software performance.

Many of the software requirements are common to all the future experiments. The goal is then to develop generic code, independant of which experiment it is to be used for, with high re-usability, high reliability, and using efficient algorithms. Making generic code allows a larger user-base which feeds back into higher quality code, while at the same time reducing the overall effort needed in development and maintenance.

Design of future detectors requires detailed simulation to find the expected performance of different options. Running experiments need detailed simulation in order to extract the physics goals. GEANT4 is the universal modern tool for detailed simulation. Both real running experiments and simulation produce “hits” - information on which detector elements had particles traversing them, and what the response was. The hits have to be analysed to extract what particles were produced in the interaction: this is known as reconstruction, and can be sub-divided into tracking for charged particles and calorimetry for both charged and neutral particles. Both simulation and reconstruction need to know the shape, position, and materials of the detector: the detector geometry. Having a single geometry model that can feed all parts of the software is a major goal: it guarantees consistency and saves effort in maintenance and modifications.

Large computing resources are needed for the complex detectors. Computing hardware develops with time, taking advantage of technological advances. Software needs to evolve to take full advantage of these improvements. The near-future major trend for computers is to go to multicore CPU's, with ever higher numbers of cores. Software needs to eveolve to take full advantage of this, with multi-threading and memory management. This will benefit both simulation and reconstruction.

These then are the central four tasks of the software workpackage: generic software for the geometry model, simulation, and reconstruction; and optimising all software to take full advantage of multicore CPU's.

WP.2-Task.1 – General purpose detector geometry description package

1. General description of the task activities

Most large experiments want the geometry described in one place. This is efficient and eases maintenance during the life of an experiment: changes are only required in one place, guaranteeing consistency everywhere else. This central geometry then needs to be interfaced to all the other software packages, which often have very different requirements on level of detail needed etc.: the optimum level of detail needs to be accessible in each package. Covering several future experiments reuires a comprehensive and flexible generic model. Methods for dealing with real detector layouts with imperfections - mainly mis-alignments and calibrations - which are time dependent also need to be developed, including databases with fast access to large data sets and efficient memory use to cache the information.

2. Organization participation

Participant acronym	RAL	DESY	CERN
Estimated person-months per participant:	18	72	30

3. Objectives

1. Develop a detector-independent geometry model, using best practice from the current packages available, and with sufficient flexibility to cope with everything needed by the future detectors
2. Optimize it for representation in memory and database to allow fast access to detector parameters, including alignment and calibration constants
3. Allow for mis-alignments of detector elements
4. Create interfaces to produce geometries for Geant4, Fluka, digitization, reconstruction, alignment, and event display programs. (Fluka is a program used to estimate backround and radiation levels at detectors and accelerators).
4. **Description of work**

3.1: Integration into ILC software for performance validation and feedback on usability.

List of Deliverables for the task

Deliverables of task 1	**Person month estimate**	**Description/title**	**Nature**[1]	**Delivery month**[2]
2.1.1		Software Design Document based on current geometry models and requirements of the detectors	R	12
2.1.3		Software package with access to database etc.	O	36

		for alignment and calibrations		
2.1.4		Interfaces to export geometries to software applications	O	48

List of Milestones for the task

Milestones	Description/title	Tasks involved	Delivery month[2]	Means of verification
2.1.1	Running prototype with limited functionality to demonstrate applicability of the model	2.1.1 and 2.1.2	18	Quantative evaluation of processing speed and memory use
2.1.2	Fully functional geometry model		36	Use in experiments
2.1.3	All interfaces to client software		48	Use in experiments

[1] Nature: R=Report, P=Prototype, D=Demonstrator, O=Other
[2] Counted from the starting date

WP.2-Task.2 – Reconstruction Software

1. General description of the task activities

The raw data (hits) from the detectors has to be reconstructed into information on the tracks of the particles that were created. The large variations between detectors and the complexity of the data mean it would be too ambitious to try to develop a unified general purpose reconstruction software package. However, many tasks within such a package are common to all detectors, such as charged particle tracking and calorimetry. A toolkit should be developed, independant of detector geometry, to ease the task of writing high-quality reliable software for these tasks.

As beam intensities increase, more interactions occur alongside the interactions of interest, known as pile-up events. These increase the amount of data to be handled. More efficient ways to handle the high particle multiplicity need to be developed.

Reconstruction needs accurate information on detector positions. This information is usually best obtained from the data itself, in a process known as alignment. High quality generic alignment software could considerably reduce the effort typically needed for detector alignment.

2. Organization participation

Participant acronym	RAL	INFN	IN2P3/CNRS	IFIC

Estimated person-months per participant:	36	24	30	30

3. Objectives

1. Develop a toolkit based on best available practice for charged-particle tracking and calorimetry.

2. Develop software to handle high particle-multiplicities: Optimise algorithms, memory management, data-base access, and file handling.

3. Develop general-purpose alignment software.

4. **Description of work**

Particle flow/calorimetry: The work would extend the clustering and particle flow calorimetry techniques developed for the ILC into a general purpose, detector independent, toolkit. The reconstruction would be interfaced to a generic description of calorimeter hits and tracking infomation based on the minimal version currently used in the ILC software framework. Calorimeter and overall detector geometry would be accessed through the geometry system developed in Task 2.1. The software will be integrated into the existing ILC software framework and its performance validated using the various ILC detector models.

List of Deliverables for the task

Deliverables of task 1	Person month estimate	Description/title	Nature[1]	Delivery month[2]
2.2.1		Software design document for tracking	R	12
2.2.2		calorimetry		
2.2.3		alignment		

List of Milestones for the task

[1] Nature: R=Report, P=Prototype, D=Demonstrator, O=Other
[2] Counted from the starting date

Milestones	Description/title	Tasks involved	Delivery month[2]	Means of verification
wp.t.1				
wp.t.2				

WP.2-Task.3 – Detailed Simulation Software Development

1. **General description of the task activities**

Development of new detector designs involves detailed simulation of the response of detectors to particle physics events. Running experiments rely heavily on detailed simulation to extract physics measurments from the data. Simulation requires large computing resources - including heavy use of GRID facillities.

Geant4 is the major package for detailed simulation. While it is mature software, in use throughout the world, improvements are still needed - particularly for the design of future detectors.

Gaseous detectors are simulated using 3 programs: Heed simulates the liberation of electrons in gases due to the passage of charged particles; Magboltz calculates the motion of electrons in gases with electric and magnetic fields; Garfield simulates the performance of drift chambers, optionally using Heed and Magboltz. These need integration with other software packages as well as maintenance and support.

2. Organization participation

Participant acronym	CERN	INFN	Blondel to say
Estimated person-months per participant:	12	48	18

3. Objectives

1. Geant4:
   a. Improved simulation of low-energy (< 1 GeV) pions and muons.
   b. Facilitate mis-alignment studies by providing tools to detect volume-overlaps

2. Pile-up: Develop a digitisation pile-up framework, independant of detector geometry, which can handle events and files efficiently. For example, the sLHC needs 400 minimum-bias events superimposed for each bunch crossing, with ~15 consecutive bunch crossings modelled. Develop digitisation tools to take into account the beam-time structure and detector response from earlier beam crossings.

3. Gaseous detectors:
   a. Support and maintenance of Heed, Magboltz and Garfield
   b. Integration of Heed into Geant4
   c. Integration of Garfield with other tools such as the particle-physics analysis package Root
   d. Further develop these packages to simulate the modern Micro-Pattern Gas Detectors (MPGD)

4. Description of work

List of Deliverables for the task
(typically 1 per task per year)

Deliverables of task 1	Person month estimate	Description/title	Nature[1]	Delivery month[2]
wp.t.1				
wp.t.2				

List of Milestones for the task

[1] Nature: R=Report, P=Prototype, D=Demonstrator, O=Other
[2] Counted from the starting date

Milestones	Description/title	Tasks involved	Delivery month[2]	Means of verification
wp.t.1				
wp.t.2				

WP.2-Task.4 – Parallelization of Software Frameworks to exploit Muti-core processors.

1. General description of the task activities

With current CPU's containing four cores and next-generation ones expecting up to 64, many particle physics applications - in simulation, reconstruction, and online-triggering - need considerable adaptation to optimise use of the extra processing power. Event parallelism can be exploited by launching as many processes (threads) as there are cores, provided thread synchronisation techniques are available. Simulation, reconstruction and triggering are all memory-intensive but it is inefficient and costly to increase the memory in line with the number or cores. Much of the memory - geometry and calibration constants for example - is common to all threads, wgich can be exploited by developing techniques for sharing memory between many cores.

Once the best techniques for thread synchronisation and memory management have been identified, implementations for the software frameworks currently used in particle physics will be developed. These include Geant4, Root, ILC reconstruction, and the ATLAS, CMS and LHCb frameworks Athena, CMSSW, and Gaudi. Adapting the framework to exploit multicore architectures will automatically benefit the applications built on them, with no - or only minimal - changes needed to specific algorithmic code.

2. Organization participation

Participant acronym	CERN	UK

Estimated person-months per participant:	96	18

3. Objectives

1. Explore thread synchronisation and memory management techniques. Select the most promising and develop them.

2. Optimise the major particle physics frameworks to run on multicore processors, using the techniques developed above.

4. Description of work

List of Deliverables for the task

Deliverables of task 1	**Person month estimate**	**Description/title**	**Nature**[1]	**Delivery month**[2]
wp.t.1				
wp.t.2				

List of Milestones for the task

[1] Nature: R=Report, P=Prototype, D=Demonstrator, O=Other
[2] Counted from the starting date

Milestones	**Description/title**	**Tasks involved**	**Delivery month**[2]	**Means of verification**
wp.t.1				
wp.t.2				