SOFTWARE TOOLS FOR ILC DETECTOR STUDIES

Roman Pöschl

Laboratoire de l'Accélérateur Linéaire (LAL)
B.P. 34 - F-91898 Orsay

This article presents a review on the main issues of the software and computing tools developed ILC Detector related studies. It works out common efforts but also differences among the efforts within the three different regions in which the detector R&D is pursued. It outlines the main features of the software packages and highlights results which were obtained by studies obtained within the frameworks. The grid is constantly evolving to be the computing environment for the studies.

1 Introduction

Software plays an important role in all aspects of the ILC detector development. Comprehensive software tools are essential to define the key parameters of a detector layout ready to achieve the goal of $30\%/\sqrt{E}$ of energy resolution. Currently, the R&D for the ILC detectors is performed within three regions comprising four different concepts for the detectors. These are namely the LDC, GLD, ALCPG and 4th Concept studies.

The computing environment as currently established for ILC Detectors comprises the core software, including the algorithms and the basic data models as well as the application of grid tools in order to perform the processing of Monte Carlo files and, in case of test beam efforts, real data [2]. In addition to that database services are provided to support the various efforts. Figure 1 shows a general overview of the ingredients of the ILC software and computing infrastructure.

Though the software frameworks differ among the three regions and four detector concepts there is a considerable effort in order to make the results interchangeable.

2 The Actual Software

The backbone of the ILC Software is the LCIO [3] package. It features a data model with well defined interfaces to common objects used in HEP studies. The application of such a data model clearly facilitates the exchange of results between different studies and therefore the comparison between detector models. Developed by SLAC and DESY IT groups, it is currently the de-facto standard for the ALCPG [4] and LDC [5] studies. Implementations of LCIO do exist for the java, C++ and Fortran programming language allowing therefore for a large community to benefit (and contribute) from (to) the existing algorithms. The SIO package is employed for data persistency and results are stored in so called *'lcio files'*. The GLD study as well as the 4th concept have developed their own root' [6] based framework but envisage to provide their results in the LCIO format in order to facilitate the interchangeability of results [7, 8]. Being at a first stage developed and designed for full detector simulation studies, LCIO is increasingly applied in test beam studies such as within the CALICE collaboration. This strategy will permit to transport easily results from

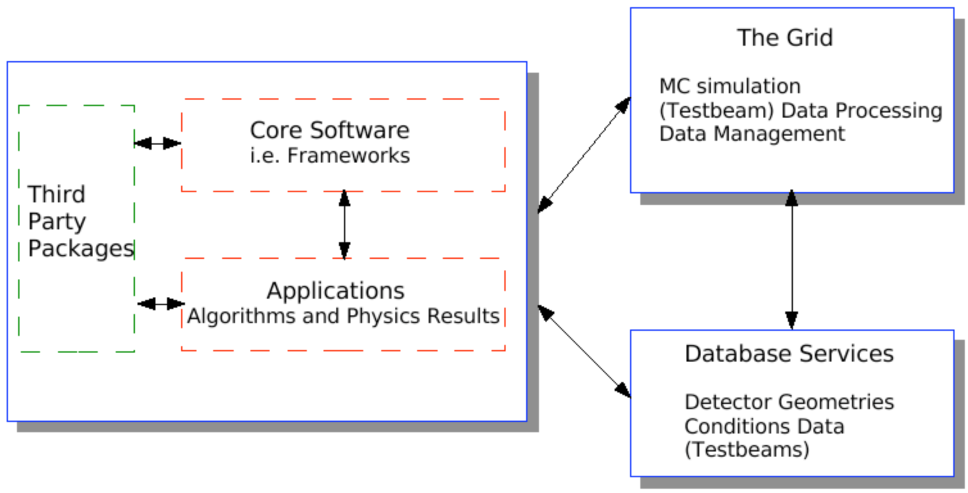

Figure 1: Ingredients of the ILC software and computing environment.

these test beam experiments into the full detector studies. Secondly, algorithms developed within the detector studies can be applied to 'real' data. The application of LCIO allows at an early stage the definition interfaces to DAQ systems, a project to be realized for the next generation of test beam efforts.

The simulation of the various detector proposals and prototypes employed in test beams is based on the GEANT4 [9] software package. In particular for several test beam efforts also GEANT3 implementations are maintained. These won't be described further here, The actual geometry is fed into GEANT4 by several methods. Within the MOKKA package [10] as used for the LDC study the descriptions are stored within a mysql data base. Within the ALCPG study the geometry is read via the package LCDD into SLIC which is the simulation package. The detailed detector description is defined within xml files. The framework allows for a rough or compact description of a given detector. The latter is transformed via a Geometry Converter into the needed xml files or other formats. Both approaches allow for a flexible adjustment of detector geometries as needed for detector optimization studies where the compact description facilitate the performance of quick studies in which the details of the detector geometry are of minor importance. Cross implementations, i.e. the implementation of one detector concept in the framework of the other concept, do exist, however on a still too low level.

The simulated files are subject to a reconstruction chain which exists for all concepts in a more or less complete form. The LDC concept uses the software package MARLIN. MARLIN provides a main program and users can implement their algorithms in form of so-called processors. The information is transported between the processors by means of an *LCEvent* object. Using MARLIN, the LDC study has developed a nearly complete event reconstruction, combining a first detector digitization, track reconstruction and vertex finding, calorimeter reconstruction jet finding and finally a particle reconstruction. Figure 2 shows the results of a recently published vertex finding suite [11] which is fully integrated into MARLIN. Table 2 gives an overview on the simulation packages and reconstruction

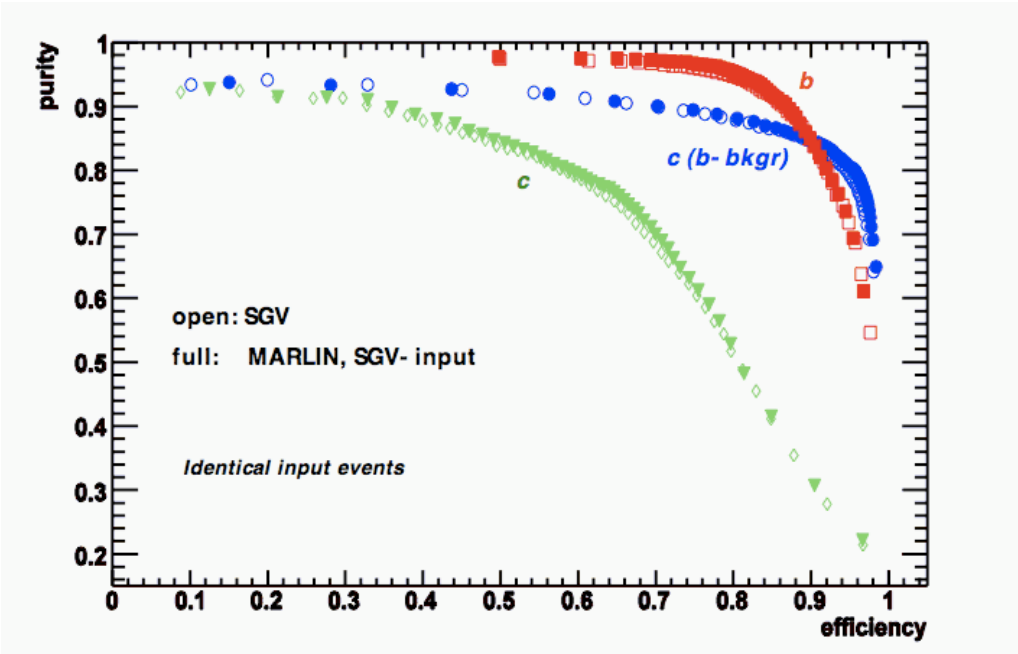

Figure 2: Flavor tagging efficiencies as obtained within the MARLIN reconstruction chain.

packages used within the four concepts

Concept	Simulation	Reconstruction	Webportal
LDC	MOKKA	MARLN	`http://ilcsoft.desy.de`
ALCPG	SLIC	org.lcsim	`http://www.lcsim.org`
GLD	JUPITER	URANUS	`http://ilcphys.kek.jp/soft`
4th Concept	ILCROOT	ILCROOT	`http://www.fisica.unile.it/~danieleb/IlcRoot/`

Table 1: Software frameworks used in the four concepts.

The ALCPG study assembles the reconstruction algorithms within the org.lcsim package. The GLD Study maintains the package URANUS which is a suite for reconstruction and analysis algorithms. An interesting approach is followed by the 4th Concept. In collaboration with the ALICE Experiment and others, the development and application of a generic reconstruction framework for HEP experiments, called HEPROOT, is under study. Such a framework would be largely based on the root system.

Having a full chain of reconstruction available allows for the application of recent Particle Flow Algorithms such as Pandora [12] and others [13, 14] under realistic conditions and hence for the optimization the of detector layout for the particle flow approach which is said to provide the precision needed for the physics studies envisaged at the ILC. Figure 3 shows the results of optimization studies done within the MARLIN framework and the URANUS framework. Both studies lead to the conclusion that a large inner calorimeter radius is more important for an optimal jet energy resolution than e.g. the magnetic field. The processor

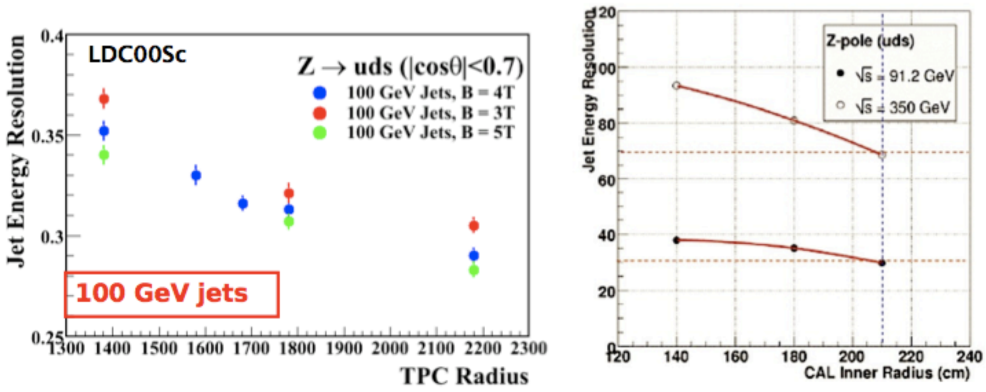

Figure 3: Examples of results of detector optimization studies within the MARLIN (left) and URANUS framework (right).

approach in MARLIN allows for an easy exchange of algorithms and therefore for detailed comparisons between different proposals.

A problem which has not yet been solved so far is the access to a given detector geometry during the reconstruction. The GEAR package is one approach to remedy this short coming. Here the MOKKA simulation outputs an xml file which can be read into MARLIN using GEAR. Under development mainly by SLAC and DESY groups, the package LCGO is foreseen to provide an interface to detector geometries independent of the software framework.

The visualisation of the results is realized by event displays. The presumably most mature approach is the WIRED Event display within the JAS suite. This package has been written and is maintained by the ALCPG study. The WIRED event display reads the detector geometry by means of HEPREP files which is, loosely spoken, a flavor of the xml language. The JAS suite allows to read in directly LCIO files with the help of suited plug-ins. By this, the information stored within the LCIO files can be conveniently coupled to the given detector geometry. Both, HEPREP files and lcio files. can be produced by the simulation and reconstruction programs of the ALCPG and the LDC study thus facilitating the exchange of results.

3 The Infrastructure

Full detector studies of tentative ILC detectors do need a significant amount of computing power to be pursued. In addition, the data and results have to be shared among the community around the world. For the ILC, grid technologies have been identified to meet these requirement [15]. The exploitation of the grid by the ILC community naturally benefits largely from the efforts undertaken for LHC computing. The virtual organisation *ilc* has been established which is hosted by DESY. Using the grid, data can be stored in a virtual file system and are accessible to all members of the virtual organisation. The ILC is supported by IT divisions in all three regions leading to a total amount of several thousand CPUs and roughly 100-200 TByte of available disk space. Since the application of grid tools is still clearly at the beginning, there is so far no dedicated organization of the computing based on the grid as it is e.g. the case for the LHC with its subdivision into TIER centers. A infrastructure like this may emerge with the forming of proto-collaborations as foreseen until the end of 2008.

Among the R&D projects for the ILC detectors the collaboration CALICE is using the grid extensively [16]. CALICE is performing R&D for the central calorimeters of the ILC detectors. For the data management and the processing of the data the vo *calice* has been established which counts currently 52 members. Up to now the collaboration has collected about 15 TByte of data. Together with reconstructed and simulated data, 30 TByte of disk space are occupied by the CALICE data.. The whole management and processing of the data is based on grid tools. The whole set of data is centrally stored at DESY but is or will be replicated to other major computing centers within the three regions. By this CALICE not only paves the way for an extended use of the grid by the ILC but delivered also important tests wrt. to a continous use of the grid for other experiments, in particular in terms of persistent data.

The rich set of parameters occurring in large scale data taking in test beam programs demands for an efficient handling of conditions data. The access to conditions data is realized by the LCCD package. It permits to store conditions data in different backends. One of these backends is a mysql database. In this case the LCCD package is itself interfaced to the CondDBMySQL [17] as written by the Lisbon Atlas group package which allows for a structured management of the conditions data. A layer and tagging tagging mechanism provides a full reproducibility of a given set of conditions data. It has to be pointed out that the current handling of conditions data is only a first attempt to establish such a software which is and will be of vital importance for any running experiment.

4 Conclusion and Outlook

All necessary parts of the software needed for ILC Detector and test beam studies do exist in a more or less mature form. Based on these tools, clear results which will influence the layout of the ILC detectors have been achieved. Emerging from different studies, the available software packages are still very heterogeneous. However, efforts are undergoing to enable the interchange of data among the studies. Here clearly the forming of proto-collaborations as foreseen until the end of 2008 will naturally lead to a larger homogenization of the software packages.

The grid has been identified as the environment for the processing and management of ILC related data. While already of vital importance for R&D projects like CALICE, it is expected that its importance for the ILC studies in general will grow considerably in the coming years.

In all fields of software the ILC community is short of manpower. This is in particular true for the development of a common and convenient event display but also for packages such as a common interface to detector geometries and for the handling of conditions data.

Acknowledgments

I would like to thank the organizers of the LCWS07 for the opportunity to give this presentation at the workshop. I would also like to express my gratitude for the marvelous organization of the complete event.

References

[1] Slides:
http://ilcagenda.linearcollider.org/contributionDisplay.py?contribId=8&sessionId=1&confId=1296.

[2] For a comprehensive overview see:
http://ilcagenda.linearcollider.org/sessionDisplay.py?sessionId=76&slotId=0&confId=1296#2007-05-31.

[3] F. Gaede *et al.*, *"LCIO - A persistency framework for linear collider simulation studies"*
LC-TOOL-2003-053, see http://www-flc.desy.de/lcnotes
See also these proceedings for an update.

[4] N. Graf *"ALCPG Software Summary"*. These procedings.

[5] For an overview on the LDC software packages see http://ilcsoft.desy.de.

[6] For a reference see http://root.cern.ch.

[7] A. Miyamoto *"Software Tools in GLD study"*.
Talk given at the ILC Software Workshop Orsay (France) May 2007
http://events.lal.in2p3.fr/conferences/ILCSoftware/

[8] C. Gatto *"Tools in 4th Concept Study"*.
Talk given at the ILC Software Workshop Orsay (France) May 2007
http://events.lal.in2p3.fr/conferences/ILCSoftware/

[9] S. Agostinelli *et al.*, *"GEANT4 - A simulation toolkit"*, NIM **A 506** (2003) 250

[10] For a reference see http://polzope.in2p3.fr:8081/MOKKA.

[11] S. Hillert *et al.*, *"The LCFI Vertex Package"*, These proceedings.

[12] M. Thomson *et al.*, *"Status of Particle Flow Reconstruction with PandoraPFA"*, These proceedings.

[13] N. Graf *et al.*, *"PFA Development in the US"*, These proceedings.

[14] T. Yoshioka *et al.*, *"GLD PFA Studies"*, These proceedings.

[15] See for example A. Gellrich, *"ILC and the grid"*, These proceedings.

[16] R. Pöschl *Calice and the grid*.
Talk given at the ILC Software Workshop Orsay (France) May 2007
http://events.lal.in2p3.fr/conferences/ILCSoftware/

[17] For a reference see *http://savannah.cern.ch/projects/conddb-mysql*