7 April 1998
David Dagenhart
Dennis Box

LHC++, Overview and Support/Maintenance Considerations

Most of the information is straight from the LHC++ web pages
See http://wwwinfo.cern.ch/asd/lhc++/index.html

LHC++ strategy

- **GEANT-4, MCLIBS++**
- **HepExplorer, IRIS Explorer***
- **HepVis, HepInventor**
- **OpenGL*, OpenInventor*, MasterSuite***
- **HepMath**
- **NAG C library***
- **CLHEP**
- **HistOOgrams**
- **HepODBMS**
- **ODBMS (Objectivity*) + persistent STL**
- **Standard C++ libraries***

*** Commercial components. The strategy is to use standard commercial components wherever possible and supply HEP extensions as required. In its evaluation, Fermilab should consider whether this and all other third party software used in CDF/D0 will successfully link.**

Schedule for the overall LHC++ system

- Working group created 1995
- There is an existing alpha/beta release. Different components are at different levels of maturity. I've actually seen a histogram created and plotted. I understand that the very newest versions will do MINUIT fitting and support ntuples. The interface and functionality still needs lots of work. This is a major concern for CDF/D0 given our schedules.
- Around June there will be another alpha/beta release. This will include full end user documentation. (Based on Explorer 3.9, OpenInventor 2.4, MasterSuite 3.0, NAG C Library Mark 4, Objectivity 5.0, ObjectSpace Internet[ToolKit] 2.0)
- First full production release scheduled for mid 1999. All components should deliver at this time, except the new LUND++ generators will probably still be in beta.

Supported Platforms

- SUN and HP are currently supported.
- NT, SGI, IBM, and DEC should be ported by the end of 1998. NT will be done first.
- LINUX schedule is unknown. In fact, it is not listed on their web site, but I have heard verbal claims that it will be supported. It cannot occur until after Objectivity is ported to LINUX. I do not know whether Explorer is supported on LINUX yet.

1998 Plan for Operating System and Compiler Versions

- SUN, Solaris 2.5, SunPro 4.1
- HP, HP/UX 10.20, aCC A.01.06/08
- NT, 4.0, MVC++5.0 using MS stdlib
- SGI, IRIX 6.4, C++ 7.1 -n32
- IBM, AIX 4.2, XIC 3.1
- DEC, Digital Unix 4.0, C++ 5.5
- LINUX, unknown

Licenses

For CERN experiments, license acquisition and maintenance will be managed centrally, including licenses at remote sites. If we followed that model, Fermilab would purchase licenses and maintenance for the experiments, both onsite and for use at remote locations. Must investigate the costs. They are significant, although CERN has apparently negotiated some good deals for themselves.

- Objectivity and ObjectSpace. ObjectSpace STL and string classes are included with Objectivity.
- Explorer
- MasterSuite
- OpenInventor
- OpenGL. These are relatively cheap and can usually be obtained bundled with the machine when it is purchased. Only run time licenses will be needed.
- NAG C Libraries
- Standard C++ Libraries
- ObjectSpace Internet[Toolkit] ? I'm not sure what this is and whether it is required for LHC++.

OpenGL

- Software interface used to create 2D and 3D graphics, with a powerful array of functions.
- It is stable. It has been around over 5 years. We probably would only need runtime licenses. We would likely take the product off the shelf and never need or want to modify it.
- Licensed OpenGL implementations must pass conformance tests and come from a single specification and language binding document. It is portable across platforms and operating systems.
- It is a standard independent of high energy physics. It is used by an impressive array of companies and institutions. Implementations (including optimized hardware implementations) are provided by a wide range of companies.
- It is well documented.

OpenInventor

- Built on top of OpenGL. Higher level toolkit for 3D graphics. Objects like cubes, text, lights, PostScript etc ...
- It is platform and operating system independent. Originally developed by SGI, and now licensed to other companies (for example TGS) for implementations on other platforms (SGI, DEC, IBM, SUN, HP, LINUX, NT, ...). Takes advantage of hardware accelerators. It is a product that has been around several years and many applications have been written on top of it.
- It is widely used outside of high energy physics.
- Established the standard format for 3D scenes (VRML).
- It is well documented.

Iris Explorer

- Built on top of OpenInventor. This is a system designed for data visualization.
- The system combines building blocks called "modules" to perform data visualization. Modules process input data, produce output data, have adjustable parameters, and perform some function. Complex data manipulations are performed by connecting the modules together. A "map" defines the way the modules are connected. One can operate this system through a GUI interface, scripts, or a command line interface.
- It is widely used outside of high energy physics, although not as widely as OpenGL and OpenInventor. It is a mature product that has been in existence since 1991. Originally developed by SGI and has been supported by NAG since 1994.
- It is portable and supported on (SGI, DEC, IBM, SUN, HP, NT, ...). I do not know about LINUX, but it not listed as supported on NAG's web site.
- It has no C++ interpreter, it dynamically links code.
- It is well documented.

Modularity

- OpenGL, OpenInventor, and Explorer appear to me to be the backbone and foundation of the system. One would have a new system if one replaced one of these components. At this level it is not modular.
- OpenInventor and Explorer are designed to be extensible and the rest of the system should be very modular. OpenInventor contains a standard set of "nodes" and "collections of nodes". It is designed so that new nodes can be created and used with system. Explorer comes with a standard set of modules, but it is assumed that the user will create new modules and maps. The user can also create new data types for use by the modules.
- Many of the other components of LHC++ are OpenInventor "nodes" or Explorer "modules". We could also create our own "nodes" and "modules". In addition, there are many "nodes" and "modules" available from sources outside of HEP.

MasterSuite

- Master Suite consists of OpenInventor extension nodes. It provides functionality for 2D and 3D charting and graphing like axes (linear, logarithmic, etc...), pie charts, bar charts, error areas, plus many more complex plotting functions than I can describe here. It provides high quality vector graphics output in several formats (Postscript, CGM, HP-GL,etc ...). It provides tools for building portable user interfaces.
- This is a commercial product provided by TGS, a California based company that specializes in 3D graphics.

HEPInventor is built on top of Master Suite. It is free and written by the LHC++ group. It provides the interface between the user data (for example, histograms) and the graphics packages. It defines the page layout of a plot (The analgous things in PAW would be zone, set, opt, igset, etc ...).
There is a class which disconnects the input data format from the page layout to make it easier to change the format of the input data. They are attempting to build modularity into the system at this level.

HepExplorer and HistOOgram

- HepExplorer is a set of Explorer modules written by the LHC++ group. They are free. It contains modules which allow the user to create, retrieve, delete, manipulate, and display histograms. There are new Explorer data types which are defined.
- There is a module which runs the MINUIT fitting package. There are other packages which are being considered for MINUIT replacements.
- HistOOgram is a class library that defines the various histogram objects. These classes are used by HepExplorer modules. The HepExplorer modules also use HepInventor to create plots.
- This is new software which is still under development, but there is a version which runs now.

GEANT-4, MCLIBS++, HepVis

- I'm not sure whether these components come under our charge. GEANT-4 is a toolkit for detector simulation. MCLIBS++ is related to monte carlo generation. HepVis is an extension to OpenInventor which would have applications towards an event or detector display. But there are a couple of important points for support/maintenance.
- Support/maintenance costs will be cheaper if these different pieces of software depend on the same underlying tools. For example, if we use OpenGL and OpenInventor for an event display, then we will already have licenses and support.
- The proposed GEANT-4 system runs on top of Objectivity. If we want to use GEANT-4 we may need to buy licenses for that as well.
- If we do not support LHC++, it will make it more difficult to use GEANT-4, and MCLIBS++.