LHC Computing Review
Answers to SPP

I - Process Planning Training and Milestones

John Harvey
March 15th, 2000

Disclaimer

Detailed answers have been written up in a technical note[1].

Please note :

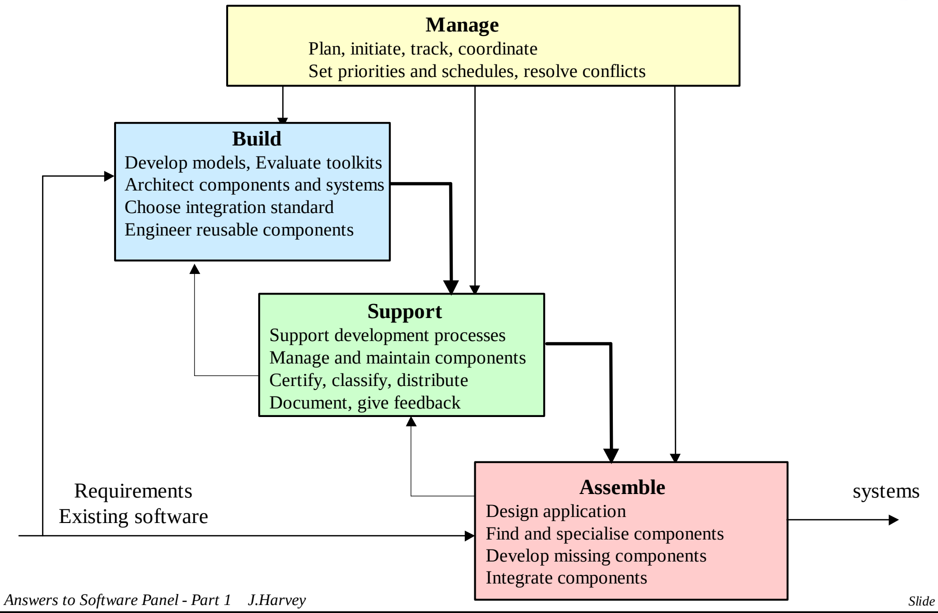
We have not had the chance of consulting widely with our colleagues and there may therefore be some factual errors that we will need to correct in due course.

1. LHCb answers to the SPP questions

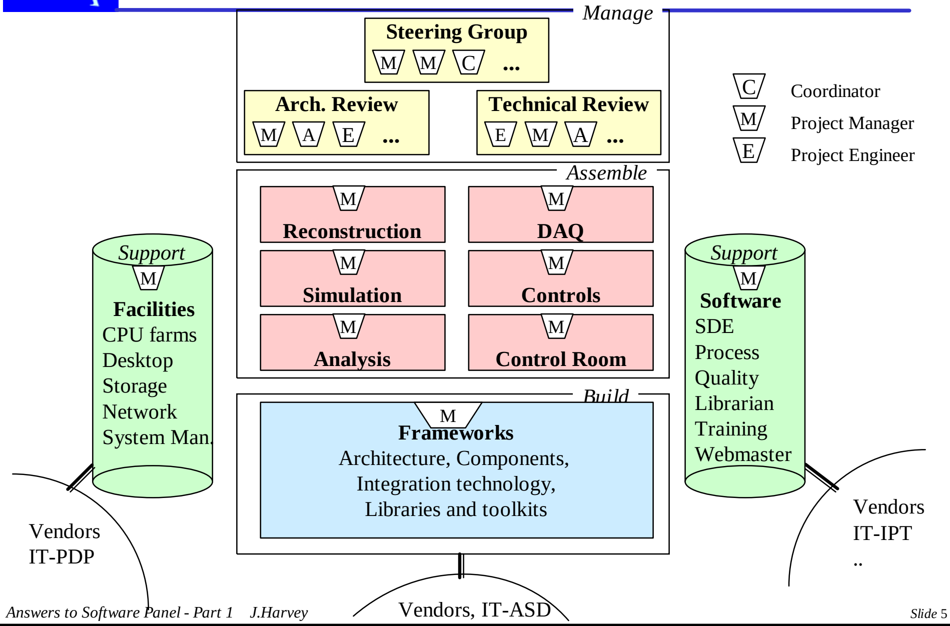
1.1 Which elements of the Computing and Software Organization participate, and interact, to effect the design and development of the software?

- Scope of the LHCb computing project covers the computing infrastructure, hardware and software, for all computing related activities in the experiment
  - data acquisition and control system (on-line)
  - data processing applications (off-line)
  - desktop computing and support of documentation
  - collaboration tools etc
- By organising all activities under one organisation we aim to minimise unnecessary duplication and make efficient use of our resources e.g.
  - between DAQ and controls
  - between online and offline (reuse of software)

1.1 Process for Organising Software Development Activities



LHCb Computing Project Organisation



Strategy for development of new software - I

- We are convinced of the importance of the architecture
- Appointed an architect with a combination of skills :
  - software engineer - designer and technologist (OO mentor)
  - physicist - knowledge of data processing applications
  - manager - form, lead and inspire the design team
  - visionary - have picture of what architecture should look like
- Start with small design team 6-8 people
  - need domain specialists experienced in design/programming
  - need librarian
- Control activities through visibility and self discipline
  - meet regularly - in the beginning every day, then twice per week, now once per week
- Collect use-cases (person dedicated), use to validate the design
- Establish the basic design criteria for the overall architecture
  - architectural style, flow of control, specification of interfaces

Strategy for development of new software - II

- Identify components, define their interfaces, relationships among them
- Build frameworks from implementations of these components
  - "framework is an artefact that guarantees architecture respected"
- Frameworks used in **all** event data processing applications:
  - high level trigger, full reconstruction, simulation, physics analysis, event display, data quality monitoring, bookkeeping, ..
- Make technology choices for implementations of first prototypes
  - language, code repository, design tool, ….

Strategy for development of new software - III

- Incremental approach to development
  - new release every few (~ 4) months
  - software workshop timed to coincide with new release
- Development cycle is user-driven
  - Users define priority of what goes in the next release
  - Ideally they use what is produced and give rapid feedback
  - Frameworks must do a lot and be easy to use
- Strategic decisions taken following thorough review (~1 /year)
- Releases accompanied by complete documentation
  - presentations, tutorials
  - URD, reference documents, user guides, examples
- Note that our process corresponds to that proposed by Jacobsen, Booch and Rumbaugh (USDP)

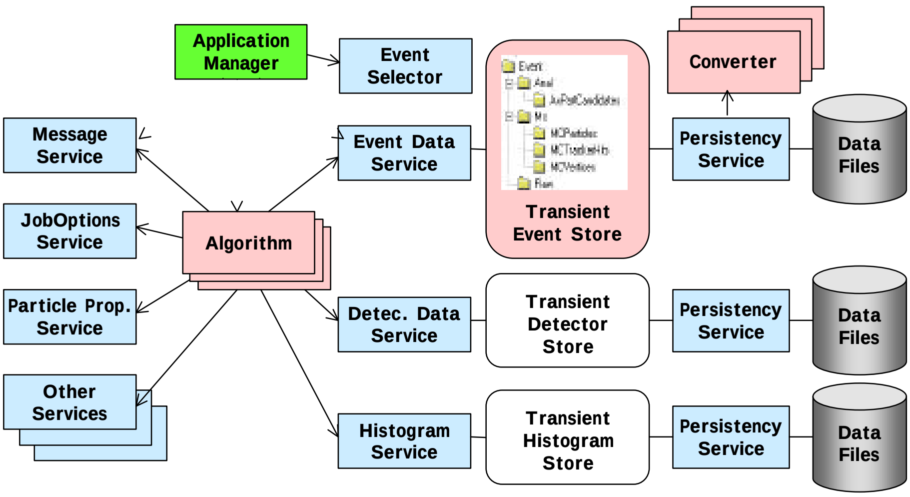
The reality

- Sept 98 - architect appointed, design team assembled
- Nov 25 '98 - 1- day architecture review
  - goals, architecture design document, URD, scenarios
  - chair, recorder, architect, external reviewers
- Feb 8 '99 - GAUDI first release
  - first software week with presentations and tutorial sessions
  - plan for second release
  - expand GAUDI team to cover new domains
- May 30 '99 - GAUDI second release
  - second software week…
  - plan for third release
  - expand GAUDI team to cover new domains
- Nov 24 '99 - GAUDI third release
  - essentially complete basic functionality
  - start to get good number of users and much feedback
- Steering group meeting once per month to track progress and plan

1.2 Which parts of the software will physicists write and which parts software engineers?

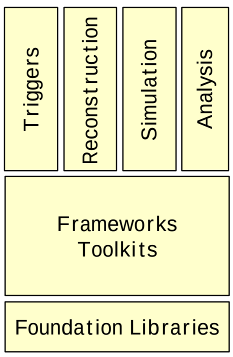
- Physicists contribute to the development of physics algorithms and those software components requiring specialist knowledge of the LHCb detector.
- Foundation libraries, frameworks and infrastructure components will be supplied by members of the computing group, who have at least some knowledge and skills in software engineering.
- Analogy - infrastucture services in the pit for installation of the detector (power, light, cooling, network, water,… .)
- GAUDI architecture defines those abstractions that physicists must supply and those services that comprise the infrastructure.

GAUDI Architecture



Software structure

- For libraries and toolkits we look to see what already exists and only roll our own if a component is missing
  - commercial
  - IT departments
  - other experiments
- A common interface model would help significantly the exchange (reuse) of software between different groups of developers



1.3 How do you stimulate and control contributions from authors spread worldwide?

- ❑ Delegate responsibility for well-defined pieces to group working remotely. Needs to have critical mass.
- ❑ In practice found that it helps significantly if someone has worked closely with the other developers at CERN and then goes back to institute and continues there
  - ↪Marseilles - main SICb author went there
  - ↪NIKHEF - maintain some presence at CERN
  - ↪RIO - no contribution yet, but will start soon
  - ↪make use of CERN associates programme

Make life as easy as possible for developers

- ❑ Monolithic software is unmaintainable
- ❑ Restructuring of SICb considerably simplified distributed development
  - ➢ originated from one person grew into one monolithic program
  - ➢ not a problem as originally all developers at CERN
  - ➢ librarian appointed, adoption of CVS and CMT
  - ➢ restructured as 35 packages, each of which can be independently released
- ❑ Provide bookkeeping utilities to make identification of data samples simple
- ❑ Grid software providing transparent access to data and cpu resources would be ideal

Good communication

- ❑ Encourage the same vocabulary
  - ➢ architecture helps to defines this through its abstractions
  - ➢ common training programme to encourage use of a particular design notation, backed up by books (LHCb library)
  - ➢ coding conventions so that code is easier to read
- ❑ Reviews
  - ➢ help to share experience and ideas
  - ➢ learn from others mistakes
  - ➢ one way to introduce mentoring
  - ➢ series of reviews of subdetector software started recently
- ❑ Documentation
  - ➢ put effort into providing user guides, reference manuals
  - ➢ project plans should be visible
  - ➢ make extensive use of web
- ❑ Software weeks and collaboration meetings ensure that people get together at least once per month

Answers to Software Panel - Part 1 J.Harvey Slide 15

1.4 What design methodology and design process does the experiment use? Why? How well does this work?

- ❑ We have not yet proscribed a formal documented LHCb software process, nor have we adopted a particular design tool. Our approach is informal and pragmatic.
- ❑ The basic design process has already been described
  - ➢ use-case driven
  - ➢ architecture-centric
  - ➢ iterative and incremental

Answers to Software Panel - Part 1 J.Harvey Slide 16

Design notation

- ❑ Learn design through specific training course that teaches UML as a modelling language
- ❑ This is backed this up through the adoption of standard software engineering texts:
  - The unified software development process: Jacobson, Booch, Rumbaugh
  - The unified modelling language user guide: Jacobson, Booch, Rumbaugh
  - Design Patterns: Gamma et al
  - Large scale C++ software design , Lakos
- ❑ Many copies of books available through LHCb computing library (2/R-008)

Design Tools

- ❑ Evaluated a number of design tools but have not identified one that is entirely suited to our needs.
- ❑ Rational Rose was evaluated, as it available at CERN.
  - ➢ Found it to be rather complex and to require a steep learning curve.
  - ➢ The code generated was found to be unreadable. In fact much effort was being put into making the design in such a way that the code was readable.
  - ➢ The general conclusion was that the tool was hindering the progress in developing the software to such an extent that it was dropped.
- ❑ Also evaluated a number of PC based design tools.
  - ➢ These tend to be simpler but are very easy to use.
  - ➢ At present we are using a tool called VisualThought which is basically a drawing tool.
- ❑ This is an area which is evolving rapidly, but we do not have the resources to do technology watch. We believe that this is an area where we could benefit from direct support from IT division.

Two questions taken together

- **1.5 How have you arrived at your workplan, or work breakdown structure? What planning process has taken place to map out the work to be done in the next two years? Who is responsible for the work breakdown structure and for keeping it up to date?**

- **1.6 What are your milestones in the area of development and how do these interact with other experiment milestones? How will you measure the success of your milestones and evaluate progress in carrying out the work plan?**

- Plans consist of outlines of major milestones for the period between now and the start of datataking
- Detailed plans exist for managing on-going software development activities

LHCb Milestones

- Magnet
  - Freeze design — Oct 1999
  - TDR tender out — Dec 1999
- Vertex
  - Design of silicon det — Jun 2000
  - TDR — Apr 2001
- ITR
  - freeze design — Jun 2001
  - TDR — Sep 2001
- OTR
  - TDR — Mar 2001
- RICH
  - complete design — Mar 2000
  - TDR — Jun 2000
- Muon
  - choose technologies — Jan 2000
  - TDR — Jan 2001

LHCb Milestones

- Calorimeter
  - Eng designs — Apr 2000
  - TDR — Jul 2000
- Trigger
  - L0/L1 TDR — Jan 2002
- DAQ
  - TDR — Jan 2002
- Computing
  - Finish first prototypes — Jul 2000
  - TDR — Jul 2002

- NB that only Magnet TDR has been submitted so far

Strategy for Software Milestones

- The period between now and the start of datataking has been divided into a number of cycles each terminated with a well-defined milestone
  - The first cycle July 1998 - July 2000 : **first prototypes**
  - The second cycle July 2000 - July 2002 : **second prototypes and TDR**
  - The third cycle July 2002 - July 2004 : **final software**
  - Th fourth cycle July 2004 - July 2005 : **integration commissioning**

First Cycle July 98 - July 00

- Represents first iteration in the production of full scale prototypes
- Demonstrate the appropriateness of basic design choices of the software architecture and validate approach to the organisation of software development activities.
- We expect to have a new reconstruction program that uses the new framework and allows new OO pattern recognition algorithms to be used in production by the summer of this year (meet milestone). This will allow us to validate physics algorithms that have been re-engineered in OO and whose performance and functionality can be compared to their FORTRAN equivalent.
- This will be closely followed by an analysis program that uses the GAUDI framework.
- After this attention will focus on integrating GAUDI with GEANT4 to produce the framework for a new simulation program (timescale not before end of 2000).

Second Cycle July 2000 - July 2002

- Investigate potential technologies for implementing the various software components
  - persistency
  - toolkits for GUI, simulation….
- Make final technology choices and prepare design specification for final software.
  - Prepare TDR July for 2002

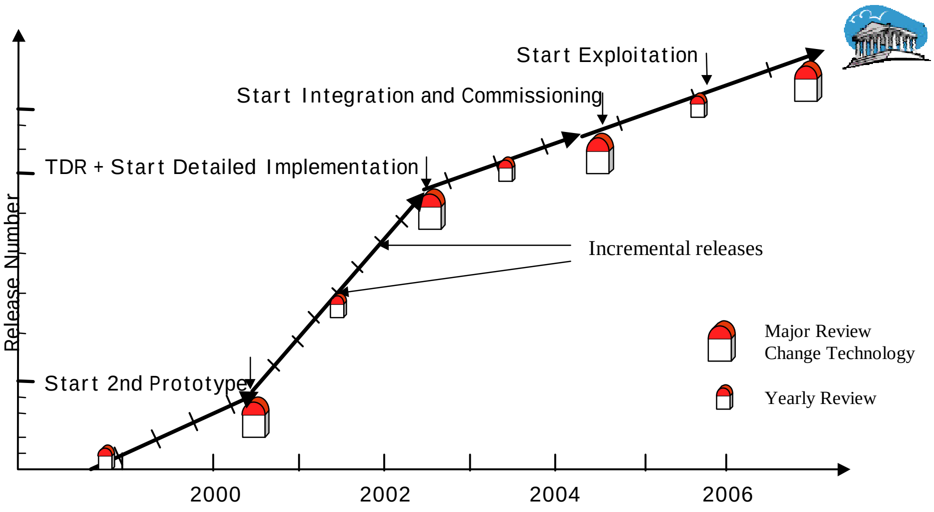
Third Cycle July 2002 - July 2004

- Produce fully functional data processing applications.
- Make large scale test of the functionality, performance and reliability of the software one year before data-taking is due to start (summer 2004).
- This software will be used in conjunction with large scale simulation tests that will also be used to test the distributed data and computing intensive nature of our applications.

Fourth Cycle July 2004 - July 2006

- The time before first datataking will be used for integration and commissioning of the complete system and for correcting and problems that have appeared.
- Efforts will be made to improve performance of cpu intensive pieces of the software
- Time of considerable investment in computing resources; cpu, storage..
- Time needed to get operational experience running large scale compute facilities

LHCb Offline Software Road Map



Answers to Software Panel - Part 1 J.Harvey Slide 27

Planning of current activities

- We have two sets of goals.
  - The first is oriented towards getting **physics** results from simulation studies that can be used in the preparation of the detector TDRs
  - The second is more **software** oriented and this is to prepare new frameworks for the main data processing applications
- Challenge is to marry the two sets of goals by carefully preparing a **migration strategy** that allow physics studies to proceed with little interference, whilst at the same time to encourage and allow new software in C++ so as not to add to the legacy code.
- A new reconstruction program (BRUNEL) is being produced. FORTRAN algorithms coexist with newly developed C++ algorithms. The planning of the migration of FORTRAN code to C++ is being discussed now. The need to make productions of simulated events for TDR preparation is a major discussion item. (more on this later)
- A detailed breakdown of tasks and responsibilities exists and progress is tracked in the weekly computing meeting.

Answers to Software Panel - Part 1 J.Harvey Slide 28

GAUDI and BRUNEL planning

- A detailed project plan has been kept using MSProject since the start of the project . It describes ongoing tasks corresponding to each release of the framework (see Appendix).
- A more detailed day to day joblist is also maintained and reviewed at the weekly meeting. This list is maintained on the GAUDI web pages.
- The development of BRUNEL is just starting and will be managed in the same way as GAUDI, with a project leader who has responsibility for obtaining and integrating software components from all subdetector developers. Attention will be given to planning tasks, identifying risks, understanding critical paths, and coordinating efforts so that timescales and deliverables are respected.

Answers to Software Panel - Part 1 J.Harvey *Slide 29*

Tracking Progress

- The success of the milestones can be measured in terms of the existence of the deliverables associated with each at the appropriate time.
- Where possible the attributes of each deliverable will be measured and compared to the requirements e.g. performance.
- The software will be continuously be used in production (simulation) to get physics results. It will be exposed to users.
- The timeliness of the delivery is easy to measure.
- Project plans will be produced and used to track progress.

Answers to Software Panel - Part 1 J.Harvey *Slide 30*

1.7 Which areas of planning and work are the highest areas of risk, in that lateness or poor quality will have far reaching affects? What is being done to mitigate these risks?

- Considering the experiment as a whole, we believe software is not on the critical path.
- Most attention in the collaboration is devoted towards making the right technology choices for the detectors and optimising the overall design

Risks : Data management

- One of the biggest challenges we face on LHC experiments is the management of the very large and complex data sets that will be produced.
- Sophisticated software will be needed to manage and access the data
- It is important that we have complete confidence in the choice of the software used to for data storage management and have sufficient control over it that we can ensure it meets all our requirements.
- Efforts to provide a solution to data have concentrated on looking for a commercial solution from the ODBMS market. Current trends in the evolution of the market and experience of some of the technical limitations of the product have been grounds for legitimate concern.
- The alternative of providing a home grown solution would also be costly in terms of development effort.

Mitigating Risks : Data management

- If a home-grown solution with ODBMS-like features is to be developed then there will need to be a significant investment in experienced manpower and sufficient time allocated.
- This is an issue which requires cooperation and agreement between all experiments to find an adequate solution that will mitigate the risk, if necessary by pooling resources. **An open debate on this issue is needed rather soon.**
- Another aspect that can help to mitigate the risk is to avoid making the software dependent on a particular persistency solution.
  - One of the basic design decisions we have taken in devising the GAUDI architecture is to separate the transient and persistent representations of the data.
  - algorithmic code has no notion of how the data are physically stored, such that and particular persistency solution can be rather easily replaced
  - e.g. at present we make use of two persistency solutions, ZEBRA, which is used for legacy FORTRAN data, and ROOT.

Risks : Quality of Trigger Software

- Stringent quality requirements must be in place for high level trigger algorithms.
- Quality can only be ensured if correct procedures are introduced to the development of the software. These include design reviews and code inspections. Data quality checks will be introduced to verify the correct functioning of the code on test data samples.
- Need to develop experience and assign resources (which we don't have)
- These checks will be applied on every new version of the software to ensure it hasn't regressed.

1.8 a)What is the plan for training? What are the various types of training required - design? use of tools? C++, other?

- Nearly all LHCb physicists programming in C++ have followed the course on C++ for Physicists given by Paul Kunz.
- In total more than 50 LHCb software developers have now followed OO analysis and design and hands on programming in C++ course
- Several have also followed specialist courses e.g. in project management, Objectivity, PVSS, ...
- Feedback from participants very positive
- Courses organised with the help of CERN's technical training dept.

1.8 b)What have you learned so far about the successes and failures of training programs and what do you intend to do in the next two years?

- Not sufficient just to learn the programming language. Any physicist wishing to do serious OO development, needs to know the basics on object orientation. All LHCb physicists working on these topics have been encouraged to follow the OO A&D course
- To be successful the training must be timely
- Three OO A&D courses organised for LHCb
- New collaborators attend the course by applying to CERN technical training department
- New specialist courses will be needed for certain software products as and when they are used (e.g. Objy, GEANT4..)
- LHCb computing group needs to develop training material for its own software (examples, workbooks etc.)

1.8 c) Do you expect that any of this be in common with other experiments?

- YES… through technical training

1.8 d) What role do you expect CERN IT to play?

- IT should provide specific training material on software developed by IT and in organising training courses on the commercial software selected for the main CERN libraries (such as Objectivity, which exists).
- In addition we foresee a special need for GEANT4 training material. Such material does exist, having been produced by GEANT4 members, largely those also working in experimental collaborations.
- This could be a very useful role for the IT departments to collect this material and help to organise training.

1.9 How will technology choices for languages, tools, database products, etc. be made? What provisions are being made for rapidly changing technology?

- We would like to track the changes in the technology and profit from the possible benefits of new technologies immediately.
- We are also aware that we need to have a periods of a certain stability
- The two wishes are contradictory and the compromise we have found is to fix the technologies (languages, tools, persistency, etc.) for a period of 2 years.
- During the general reviews of the computing project (scheduled at 2 year period) we could be fixing the choices for the next period.
- Decisions must be prepared ahead by R&D and prototyping work

1.10 What plans do you have for the long-term support of your software?

- Procedures defined for configuration management
  - CVS for code repository and CMT for building releases.
  - We are working on automatic build procedures.
  - Manpower assigned for librarian and an assistant
  - Manpower not yet available for certification of software, quality control
- Cope with turnover of collaborators
  - document requirements, designs, code, and test procedures. Where possible (semi) automatic procedures should be used to produce this material to ensure that it is kept up-to-date and this implies extensive use of software tools.
  - We have put some considerable effort into documenting GAUDI. Software reference manuals are produced automatically using a tool called ObjectOutline
  - Reviews generate a lot of useful material, documentation is produced before-hand and the results of the review must also be documented.
- Training material needs to be produced for each step. Workbooks seem to offer the most convenient format.

1.11 What quality assurance and control mechanisms are being put in place, and in which stages of the design, implementation and testing processes?

- At present we do not have sufficient manpower resources to put in place a proper software quality process. Areas where we have made some progress are...
  - design reviews - we have made reviews of GAUDI and have started this month a series of reviews on subdetector software (calorimetry, tracking and RICH)
  - coding conventions - we have established a coding conventions guidelines document. We have not yet put in production the automatic checking of code checked into the repository. We are awaiting the outcome of a project started by IT/API group
  - data quality monitoring - following each new release of SICb the production team checks the output against a standard set of histograms that represent the understood behaviour of the program. This represents a simple regression test. We expect to apply such data quality checks on all future versions of our data processing software.

1.12 What decisions on software technology and implementation choices have to be taken in the future and when do you plan to take them?

- Technology choices will be reviewed every 2 years.
  - we are currently using UML as a design notation and C++ as an implementation language.
  - We are also putting some effort (work of a technical student) to evaluate other languages (Java) and at the next major review we will debate the advantages of introducing Java.
- Choices still need to be made for some of the basic toolkits used for implementing framework services. The example of the persistency service has already been mentioned. For the simulation toolkit we are starting to get experience with GEANT4.
- The technologies to be used for the development of the software to be run in 2005 will be defined in the software TDR (expected 07/2002).

1.13 What is the required number of people contributing to the software, what is the break-up between physicists and software engineers? What is the evolution over time to meet the milestones (manpower profile)?

Activity	Need	Have	Miss	Type
Software frameworks	12	7	5	E/P
Software support	5	2	3	E
Muon software	6	6	0	P/E
Muon trigger	2	2	0	P
Tracking	6	6	0	P
Vertex	6			P/E
Trigger (L0,L1,L2,L3)	7	3	4	E/P
Calorimeter (ECAL,HCAL,PREShower)	8	8	0	P
RICH				
Total	52	34+	12+	

Answers to Software Panel - Part 1 J.Harvey Slide 43

1.14 What is the recruiting model for manpower?

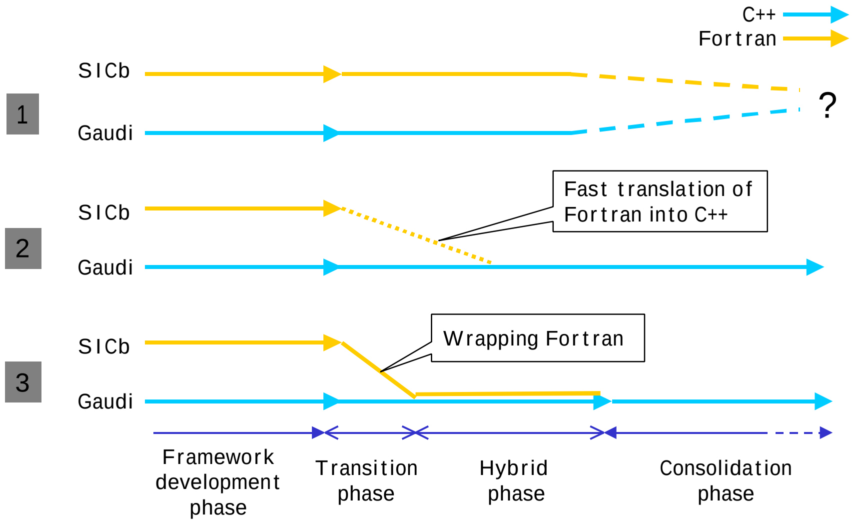
- For computing we need to recruit people with a software profile.
- Recent priorities - computer scientist to help with system management, trouble shooting problems on desktop machines and on facilities used by collaborators at CERN.
- Priority will be to secure funds for obtaining services from the CERN desktop contract.
- The highest priority for the next position in LHCb is for a physicist with software experience to work firstly on studies of the impact of background radiation. However this person would also work on the analysis framework.
- A significant amount of effort is to be acquired through the students and fellows programme at CERN. Our first priority is to consolidate the development of the online system by acquiring an Applied Fellow, which we hope to do in the summer.

Answers to Software Panel - Part 1 J.Harvey Slide 44

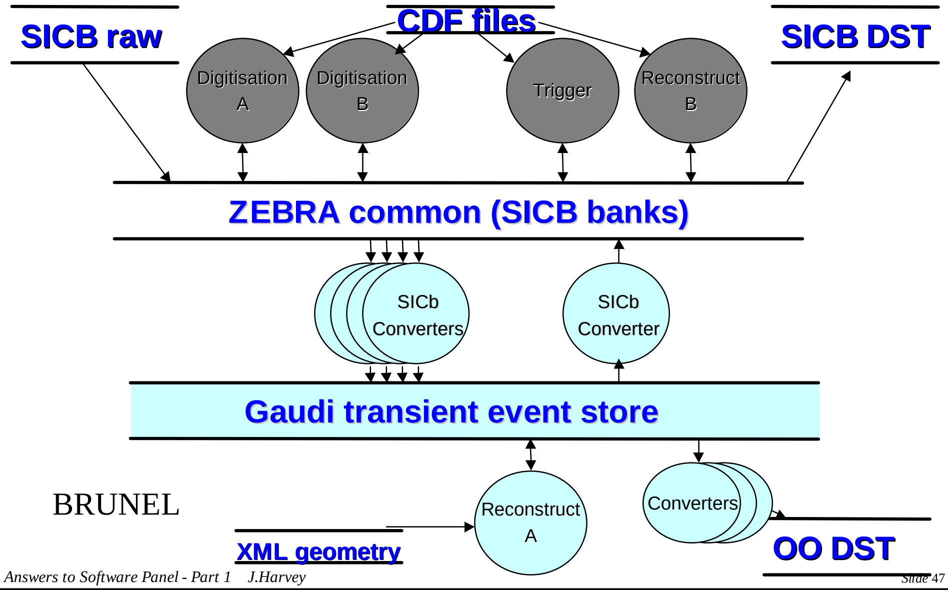
1.15 How do you plan to provide working software and do development at the same time? How do you plan to transition from existing software to final production software

- Urgent need to be able to run new tracking pattern recognition algorithms, which had been written in C++, with standard FORTRAN algorithms in production and in time to produce useful results for the detector TDR's
- Practical software goal, namely to allow software developers to become familiar with GAUDI and to encourage the development of new software algorithms in C++

Migration Strategy



Step 3- BRUNEL structure



Steps in migration

- Step 1 in the procedure involves restructuring the existing FORTRAN into its simulation (called SICbMC) and reconstruction (SICbREC) components.
  - Done
- Step 2 is to wrap digitization and reconstruction Fortran modules in GAUDI.
  - The nett result is a new reconstruction program which we call BRUNEL.
  - Expected by Easter 2000.
- In Step 3 the FORTRAN algorithms are gradually replaced one by one with new OO algorithms that use all services and features of the OO framework (event model, detector description etc.).
  - This is the hybrid phase and the aim is to keep it as short as possible as during this time FORTRAN and OO representations of components, such as the detector description will have to be maintained.
  - Start replacing FORTRAN modules with C++ equivalent.
  - Detailed schedule expected from subdetector groups at next software week (April 5-7)

1.16 Given that the support from CERN/IT is limited, how do you identify the areas where you would most like to see strong CERN/IT involvement and support? What are the arguments for central CERN support?

- We would like to see (not necessarily in priority order):
  - Support for HEP toolkits like GEANT4 - good uuser support service
  - Solution for persistency, transparent data access and storage
  - Support for foundation libraries, GUI, MINUIT, particle properties,
  - guidelines and support for "organisation, methods and tools", for documentation and information management.
  - technology tracking on tools, manage company contacts, handling licence agreements etc.
  - items not strictly software, such as tools for control and operation of compute farms, management of grid computing etc.
  - in the area of online, we rely on the controls group to supply software via the JCOP project
- Arguments for are optimisation of resources, centralisation of contacts with industry etc.
- The model for managing IT based software projects needs to be reviewed to ensure that whatever is produced will be used.