U.S. ATLAS Physics and Software Development Work Plan

Changes relative to Fri 11am version:

- *Adjusted apportionment of request between data management and control/framework; .5FTE shifted from data management to control/framework (Chris Day added) (also in PMP)*
- *David Quarrie's job description added in 2.1.5.1*
- *U Michigan database software professional added to request (also in PMP)*
- *Core milestones summary section 2.4 added*
- *DOE 2000 added to collaborative tools*
- *Several subsystem updates (muon CSCs, neutron backgrounds, …)*

This document addresses the activities and plans of the U.S. ATLAS group in physics as it relates to software and in offline software development for the ATLAS detector. ATLAS offline software is broadly divided into two categories, **core software** and **subdetector specific software**. Core software comprises the common operational framework, infrastructure, and utilities that are not specific to any piece of the detector. Detector specific software comprises detector simulation, reconstruction and calibration codes and the detector specific components of the detector description and database. U.S. ATLAS is participating in the development of select components of both core and detector-specific software and is undertaking leadership roles in aspects of both.

Core software responsibilities in the U.S. are currently the subject of well advanced negotiations with ATLAS Computing management, which is itself emerging from a period of reorganization. The U.S. groups have sought to focus on Control/Framework software and Data Management software, including work on the Event Model and it has become clear that these will be the major U.S. responsibilities. These areas are well matched to the interests and expertise of U.S. collaborators; they are closely related; and they secure for the U.S. a central role in the core software effort.

In subdetector software U.S. efforts are focused in areas which complement our hardware responsibilities. Specific software responsibilities are negotiated within the subdetector software organizations of ATLAS. Throughout the U.S. program we plan to closely connect core software development with subdetector software efforts, with the latter providing real-world testbeds for the evolving core software.

The document includes our request for FY2000 U.S. ATLAS project funding for software development. Supporting this request are detailed work plans including milestones with deliverables and personnel breakdowns. The limited availability of funding in FY2000 has led us to tightly focus our funding request on the most essential needs of our principal activities in Control/Framework software and Data Management software.

1 Physics Activities and Plans (WBS 2.1)

The main goal of the ATLAS experiment is to determine the underlying mechanism responsible for electroweak symmetry breaking and make detailed measurements of the properties of particles connected to it. This goal entails the search for the various new particles predicted in the various theoretical models that aim to explain electroweak symmetry breaking. In addition ATLAS will perform many detailed measurements involving top and bottom quarks and the other particles of the standard model. In the era before data is available, there are four main areas of physics related activity. First, strategies need to be developed to uncover new physics and simulations undertaken to evaluate how the detector can best be used to extract physics. Second, the impact of possible detector changes needs to be evaluated. The physics tools such as Monte-Carlo event generators need to be validated, made available and integrated into the ATLAS software. Finally the various physics studies provide the opportunity to exercise, evaluate and debug the complex software currently under development.

A particular example of the latter is "mock data challenges" which involve the production of large amounts of simulated data. The first of these is expected to take place in 2003 and we have planned to have sufficient CPU and disk space at the U.S. Regional center so that U.S. physicists can participate in this effort.

As well as offering the opportunity for physicists to become familiar with the ATLAS software in preparation for real data analysis, these mock data challenges also provide testing ground for the way in which the regional centers will interact with each other, with users and with CERN.

Almost all of the Physics effort is expected to be done by physicists who are not supported directly by project funds. However, we anticipate the need for a person to contribute to the maintenance and development of the Monte-Carlo generators as part of the overall ATLAS effort. One of the multi-purpose generators is written by Paige and Protopopescu (BNL) and Hinchliffe is convener of the ATLAS wide group charged with the support of these generators in the collaboration

The recently completed Detector and Physics Performance Technical Design Report (CERN/LHCC 99-14) had major participation from U.S. ATLAS physicists. Ian Hinchliffe (LBNL) was one of the two overall editors and also was responsible for the chapter on "Other physics beyond the Standard Model" (Chapter 19). Frank Paige (BNL) was coauthor of the "Supersymmetry" chapter (Chapter 20) and John Parsons (Columbia U.) was coauthor of the "Heavy quarks and leptons" Chapter (Chapter 18). Among specific contributions by U.S. members were the papers of Hinchliffe and Paige[1] on various aspects of supersymmetry. In particular these works addressed the strategies that could be used at the LHC for making detailed measurements of the properties of supersymmetric particles and disentangling the underlying model. One of the signals discussed is that arising from a neutral particle that decays to a photon and another neutral particle with a lifetime long enough so that the photon does not appear to come from the primary interaction point. While the ATLAS electromagnetic calorimeter is provides very good direction information for photons that come from the primary vertex, its performance for such "non-pointing" photons was not known. A study was carried out by Rajogopalan (BNL), Parsons, Borissov, Leltchouk (Columbia U.) and Paige[2] that showed that these photons can be reconstructed and measured with good efficiency. Other detailed studies have involved the reconstruction of tau leptons produced in supersymmetry events and decaying into hadronic final states (Coadou, Hinchliffe, Lozano-Bahilo, Loveridge and Shapiro, LBNL)[3]. Parsons and collaborators[4] investigated the detection of rare top quark decays and the sensitivity to new particles that decay into final states of a top and antitop quark pair. Shank (Boston U.) and collaborators[5] studied the performance of the muon system in the context of the decays of heavy particles.

U.S ATLAS members with current responsibilities in the physics groups are, John Parsons (Columbia, co-convener of the top working group), Frank Paige (BNL, co-convener of the supersymmetry working group), and Ian Hinchliffe (LBNL, convener of the Monte-Carlo generators group).

---

[1] I. Hinchliffe, F.E. Paige, E. Nagy, M.D. Shapiro, J. Soderqvist, and W. Yao, `Precision SUSY measurements at LHC: Point 3', ATLAS Internal Note ATL-PHYS-97-109 (1997); H. Bachacou, I. Hinchliffe and F.E. Paige, `Measurements of masses in SUGRA models at LHC', Atlas Internal Note ATL-COM-PHYS-99-017; I. Hinchliffe and F.E. Paige, `Measurements for SUGRA Models with Large at LHC', ATL-COM-PHYS-99-018; I. Hinchliffe, F.E. Paige, M.D. Shapiro, J. Soderqvist, and W. Yao, Phys. Rev. D55 (1997) 5520; I. Hinchliffe and F.E. Paige, `Measurements in Gauge Mediated SUSY Breaking Models at the LHC', hep-ph/9812233, ATL-PHYS-98-134

[2] L. Borissov, M. Leltchouk, F. Paige, J. Parsons, S. Rajogopalan, 'Study of non-pointing photon signatures of Gauge Mediated SUSY Breaking Models', Atlas Internal Note ATL-COM-PHYS-99-037 (1999)

[3] Y. Coadou, I. Hinchliffe, J. Lozano-Bahilo, L.C. Loveridge, and M.D. Shapiro, `Identification of hadronic Tau decays in ATLAS', ATLAS Internal Note ATL-PHYS-98-126

[4] J. Dodd, S. McGrath and J. Parsons, 'Study of ATLAS sensitivity to the flavour changing neutral current decay t to Z q', ATL-COM.PHYS-99-039; N. Cartiglia and J.Parsons ' Study of ATLAS sensitivity to a heavy resonance decaying to top anti-top', ATL-COM-PHYS-99-038

[5] J. Shank *et.al* 'Studies on A,Z-prime and W-prime with the ATLAS muon detector', ATL-MUON-97-161

2 Core Software (WBS 2.2.1)

The core software provides the operating environment for the software modules written by physicists, and supplies mechanisms for control of those modules, for data input/output to them, and for communication and coordination between modules written by different people and groups. Core software also encompasses data management, including event data storage and access, and detector description, calibration, and other database-resident data. It provides functionality that facilitates reconstruction and physics analysis tasks, such as histogramming and visualization tools, and insulates the physicist user from the details of implementation and environment.

Software development in the core areas is both crucial to the work of ATLAS physicists and presently drastically under-staffed. Because it provides the environment into which physicists must integrate their own code, it must be developed and made operational early in the software process. The design and development effort for core software must, therefore, be heavily front-loaded. It is not unreasonable to expect that the next several years will require more work on core software than on detector-specific and physics software. The importance of early core work is recognized by ATLAS and motivates the urgency given to defining and implementing a core software architecture for ATLAS through the design and specification work of the ATLAS Architecture Task Force and the forthcoming implementation work of the emerging Architecture Team. The recognition that a high proportion of the core software expertise available to ATLAS is found in the U.S. is reflected in the major roles offered to U.S. collaborators in core domains, as discussed below.

The scale and complexity of the offline software required for timely and effective extraction of physics in ATLAS make it essential that modern software technologies and methodologies be employed in order to make the computing task tractable at reasonable manpower levels and on the required schedule. The scale and sophistication of the required software further dictate that dedicated computing professionals participate in all aspects of software development. A professional software engineering effort, closely partnered with physicists, is required. Core software will be particularly dependent on computing professionals in the design and development effort because of the highly technical content and stringent quality and reliability requirements in the core domains.

Thus both schedule and professionally demanding requirements on the software lead us to include software professionals in our manpower needs to support and complement the work of physicists in software development, particularly in core software domains. We expect many of these professionals will have physicist training – beyond the obvious advantages of familiarity with the field, we are best able to hire and retain personnel with an interest in the scientific program – but the degree of commitment and specialization and level of professionalism in software required of such positions precludes research physicists from filling these roles.

Based on current negotiations with ATLAS we anticipate the U.S. playing a leading role in the control/framework software domain within ATLAS, with Craig Tull (LBNL) leading the U.S. effort. We have already established a leading role in the data management/database domain with the appointment of David Malon (ANL) as co-leader of the ATLAS database effort, and we propose to assume responsibility for approximately half of the ATLAS-wide task in this domain. Within the U.S. organization, Tull and Malon are the project managers for the respective core sub-projects. Subsequent sections describe the U.S. program in these domains.

2.1 Control/Framework Software (WBS 2.2.1.1)

ATLAS is committed to the development of object-oriented (OO) software grounded in accepted international standards and practices. The long lifetime of the experiment, the complexity of the required software and the distributed developer environment (we expect ~85% of the software effort to be outside of CERN) all argue for a highly modular system with well defined interfaces.

ATLAS has addressed the issue of software design in its Computing Technical Proposal, and more recently in the Architecture Task Force Report pertaining to core software. An ATLAS software development methodology has been codified in an ATLAS Software Process (ASP) including formal design reviews and required stages of documentation. Details of the ASP are currently under review, with the aim of streamlining the process; we expect, however, a continued emphasis on modularity, maintainability and documentation.

Control/Framework software provides the operating environment for the software modules written by physicists, and supplies mechanisms for control of those modules, for data input/output to them, and for communication and coordination between modules written by different people and groups. It insulates the physicist user from the underlying details of the operating system, data formats, and communication protocols.

2.1.1 Architecture Planning

For the duration of the recently concluded ATLAS Architecture Task Force (July 1, 1999 to October 30, 1999) D. Quarrie and M. Shapiro devoted the whole of their ATLAS time to its activities, assisted by C. Tull and others from LBNL. The principal result of the ATF is a report detailing decisions relating to the overall architecture of ATLAS software. The ATF also decided to appoint an ATLAS Architecture Team to follow up on its work by pursuing the detailed design and implementation of the ATLAS software architecture, with the aggressive initial milestone of delivery in May 2000 of a first prototype framework.

As of December 9 1999, P.Calafiura, D.Quarrie, and C.Tull have been appointed to the emerging Architecture Team. This team has no finite term and will be meeting to discuss and develop the ATLAS software architecture for the forseeable future.

2.1.2 Control/Frameworks and Components

As stated above, the complexity and sheer volume of code that is necessary to support and accomplish the event reconstruction, data mining, and data analysis for an experiment like ATLAS presents a tremendous challenge for developers of the software infrastructure in general, and of the control software domain in particular. Other experiments have faced this type of challenge (albeit at a smaller scale) before and have addressed it in different ways.

One approach which has been successfully tried in several High Energy and Nuclear Physics (HENP) experiments, but has been perfected in none, is the use of an analysis framework. An analysis framework is a software environment into which software contributions written by many authors can be integrated in a way which allows and eases integration and inter-operation of those software contributions. The framework concept is well known in the Information Technologies (IT) community and has been described many ways. Gamma, et al., in the Design Patterns[6] book distinguish between a toolkit (such as a class library) and a framework as follows:

> Gamma, et al., Design Patterns: "When you use a toolkit, you write the main body of the application and call the code you want to reuse. When you use a framework, you reuse the main body and write the code it calls... Not only can you build applications faster as a result, but the applications have similar structures. They are easier to maintain, and they seem more consistent to their users. On the other hand, you lose some creative freedom, since many design decisions have been made for you."

Data analysis framework-"like" programs exist and have been used in HENP experiments. However, often no distinction is made between software packages which address very different analysis-related tasks (e.g. execution control, data I/O, graphics presentation, data analysis, etc.). This tendency arises from the necessity that these tasks be seamlessly integrated in the final system. However, the resultant lack of compartmentalization complicates software maintenance and upgrades[7] and imposes a barrier to understanding the overall architecture for new software developers and users.

A recognized approach to this kind of problem is the adoption of software components as the fundamental building block of the overall software system. Again, component technology is a well known and increasingly popular approach in the IT community (CORBA, DCOM, and Java Beans are examples of component technologies that are gaining acceptance across a wide range of software industries.).

Although no formal definition of a component is accepted by the Computing Sciences (CS), there is usually large overlap between different experts' individual descriptions.

---

[6] Design Patterns : Elements of Reusable Object-Oriented Software (Addison-Wesley Professional Computing) by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Grady Booch (Designer) 1 edition (October 1995) Addison-Wesley Pub Co; ISBN: 0201633612

[7] "the size of the component to be changed has a much larger impact on effort than the size of the change itself."- Nisink, Predicting Maintenance Effort with Function Points, IEEE International Conference on Software Maintenance, Bari, Italy, October 1-3, 1997.

> Lakos, Large-Scale C++ Software Design: "A component is not a class and vice versa. Conceptually, a component embodies a subset of the logical design that makes sense to exist as an independent, cohesive unit. A component bundles a manageable amount of cohesive functionality that often spans several logical entities... and can be lifted as a single unit from one system and reused effectively in another system without having to rewrite any code."

Lakos also points out that a component-based architecture can (if properly designed) help control the physical design (as opposed to the logical design) of a large system, reducing many of the compile-time and link-time problems which can arise in any large software system.

An analysis framework recently developed by the LHCb collaboration (GAUDI) has many of the characteristics that appeal to framework experts at LBNL as well as elsewhere in ATLAS. These include an explicit attention to package dependencies, a coherent and consistent architectural approach to the design of the framework, and a formalized approach to the description and documentation of the framework design.

2.1.2.1 An Extensible, Component-Based Physics Analysis Framework

We are proposing in the context of the ATLAS Architecture Team to design, develop, and deploy an extensible, component-based physics analysis framework for use within the ATLAS experiment initially based upon the current GAUDI architecture. We believe that the resultant system will be easily usable by other experiments, and potentially even other disciplines. We do not yet propose an official or formal collaboration with the LHCb framework developers, but rather an experimental cooperation in the short term with the possibility of engaging in a formal collaboration at some future date.

Thanks to a truly unique pool of physicists and computer scientists at LBNL who have played major roles in the design and implementation of analysis frameworks for BaBar, CDF, CLEO-III, PHENIX and STAR experiments, we are in a position to solve the computing challenges created by the software complexity of the next generation of HENP experiments.

We have conducted extensive evaluations of previous ATLAS control and framework structures, along with a "market survey" of comparable systems from other experiments as a necessary first step in the development of a long-term strategy for the ATLAS control framework.

During the course of the ATF we have had presentations at LBNL from experts on Object Network Component Model (ATLAS), StAF (STAR, PHENIX, E896), CLEO III Framework (CLEO), AC++ (BaBar, CDF), and Gaudi (LHCb), and have begun their evaluation. We also built the single most complex object network configuration to date using the Object Network Component Model in an effort to investigate feasibility of using this approach to software execution and data flow control in an HENP experiment and have reported on our evalutations to the ATF. These evalutations convince us that the GAUDI framework forms the best starting point for development of a modern, object oriented analysis framework for ATLAS.

2.1.2.2 Control Framework Functionality

Because two of the primary tasks of this project are to assess the requirements for the control framework and to design an architecture which satisfies those requirments, we cannot yet with certainty describe the final form and functionality of the control framework. However, from prior experience with similar systems in other HENP experiments, we can discuss in some detail many of the necessary control framework functional requirements and potential design and implementation candidates.

Some of the primary tasks of the proposed analysis and control framework are to control the flow of the event reconstruction jobs, and the I/O of event, calibration and control data, efficiently using the scarce CPU and I/O resources available for the production batch jobs. To process each event, analysis and service modules (including I/O modules) will be connected in multiple execution paths, corresponding to different event classifications from the experiment trigger. Each path will have independent filters and I/O modules. The framework will make sure that modules common to more than one path will be executed only once. The framework will control the transitions of the modules through a finite, configurable set of states (start-up, run begin, event begin, etc.).

To manage batch jobs running for weeks on hundreds of CPUs we will provide active error handling/recovery, execution checkpoint/restart, and, if the event store is based on an ooDBMS, we intend to support ooDBMS transactions with rollback for module communications and I/O. Another very important feature for the production and on-line processing management is a journaling system capable of storing in a flat file or database the configuration of the production job: the active paths, the input databases, the version of each module in use, the parameters defining its behavior and any other information necessary to reproduce and understand the result of a production job.

Initially, the framework will be used by the core software team of each experiment (50-100 software engineers and physicists) as a development tool. As the experiments get closer to data-taking the number of "end-users", many of whom will actually contribute substantial amounts of code to the plug-in modules, will ramp-up to several hundred. The physicist who is developing a new track fitting algorithm on his/her personal computer, will typically pick a few selected events from the main event store, will reconstruct them using the new and old algorithms and will analyze the resulting tracks using an event display program and her/his preferred physics analysis package.

This has important implications for the design of the framework. It must be possible to run with only a lightweight "kernel" and a subset of plug-in modules relevant to the developer. The I/O management must be flexible enough to fetch the necessary data and calibration from the remote event store or read a subset of them from local disk and provide the plug-in modules with the same interface to the data no matter what the format or physical location. The framework must provide the user with an interactive shell offering the full functionality of the production jobs (namely the ability of defines execution paths and to configure modules), dynamic loading of modules, the ability to stop event analysis in the middle of an execution path on the basis of the information provided by an event selection or filter module. It must offer a consistent interface to pass module output data to the event display and to any of the supported physics analysis packages.

Other expected use patterns of the framework include the detector calibration and on-line monitoring, where high-speed interactive access and processing of moderate sized data samples must be granted, and Monte Carlo simulation with limited I/O and interactivity but high use of CPU resources. In these use domains, the module call overhead must be small allowing the system to run under many different situations.

To accommodate legacy and special-purpose code (libraries, graphics, etc.) as well as language evolution the framework will need to be able to bind multi-language plug-in modules. It is likely and desirable that the framework itself will be written in a single language but we can not exclude a priori the concept of a mixed-language framework.

We believe that it is essential to provide the physicist with the ability to analyze and visualize the data in the most productive manner possible for that individual. We envision the use of a format independent data conduit that will permit the use of many different analysis/visualization packages, and that can be expanded in scope as new products become available. It will offer the ability to deal with complex objects as well as simple literals, and interface directly with the user's analysis code. It is not intended to be a replacement for the likes of PAW and ROOT, but rather an interface layer between analysis code and a visualization package, permitting the user to choose whichever final format is desired. By calling on routines provided by this package, the user will be able to histogram simple variables, object members, and even entire objects, in multiple dimensions. Upon completion, the histogrammed data will be saved in whichever format was selected by the user, allowing continued analysis or visualization at their discretion.

The additional burden that will be placed on the user for the incorporation of this conduit will be minimal. It will automatically handle standard data types, and require but brief descriptions of complex, user defined objects. In the event that the user wishes to create a custom output format, skeleton programs will be automatically generated for the user to flesh out.

The use of such a data conduit will free physicists to pursue analyses in less rigid fashions than are currently in vogue. By facilitating such freedom, we can start to make industry standard tools more accessible to the average physicist, as well as allowing specialized tools to be developed for particular applications. It has often been the case that large applications which are designed to handle all possible scenarios, such as PAW and ROOT (both popular analysis packages currently in use in HENP), are by nature subject to severe limitations. By breaking this chain, and allowing freedom of choice, we hope to add flexibility that has, until now, been sorely lacking from the HENP domain.

Based upon the recommendations of the ATLAS ATF and in consultation with the newly forming ATLAS A-Team, we believe that the framework should include the following functionality.

- **Framework Manager:** Responsibility for building and configuring the application.
- **Application Manager:** Controls the event loop, driving the modules through their execution.
- **Job Options Service:** Configures adjustable parameters
- **Event Input:** Provides source of events for the event loop
- **Event Output:** Outputs event data to the persistent store
- **Data Item Selector:** Selects information within an event for output

- **Event Collection Manager:** Selects sets of event for input by the Event Input Component
- **Event Merge:** Merges information from several events (e.g. background mixing)
- **Module Interface:** Defines the interface for all major algorithmic components that are controlled by the Application Manager.
- **Transient Event Store:** Used to cache data objects during execution of the modules.
- **Event Data Service:** Provides access to data objects in the transient event store
- **Event Persistency Service:** Provides access to persistent data in the persistent event store
- **Event Data Converters:** Provides conversion between different representations of data - e.g. transient/persistent.
- **Detector Description Service:** Describes the geometry, materials and readout information for detector components.
- **Conditions Data Service:** Provides storage of time-interval based information such as calibrations, slow controls and alignment.
- **Statistics Data Service:** Histograms, ntuples and other statistics information that are collated during the processing job.
- **Magnetic Field:** A component of the Conditions Data Service (12), that is highlighted because of it's importance.
- **User Interface:** The service that the user interacts with, passing information through the Job Options Service.
- **Message Service:** Responsible for passing messages created by other components to the outside world.
- **Bookkeeping Service:** Responsible for keeping job statistics
- **History Service:** Stores the configuration information for the job such that it may be retrieved or examined after the job has completed.
- **Particle Properties:** A service that provides access to the properties of standard elementary particles.
- **Framework Utilities and Tools:** Applications and scripts to ease migration to and use of the framework.

2.1.3 Timescale and Milestones

Major milestones are foreseen as being:

- May 2000 Prototype release of reconstruction framework
- Jun 2000 Alpha release design review
- Sep 2000 Alpha release of control framework (basic functionality)
- Mar 2001 Freeze beta architecture and database interface
- Jul 2001 Full function release design review
- Oct 2001 Full function release of control framework (general use)
- Apr 2002 Freeze distributed architecture
- Jul 2002 Control framework V1 design review
- Oct 2002 Control framework first production release
- Jan 2001 Control framework V2 design review
- May 2004 Control framework second production release (post-MDC)
    - [Iteration on design reviews and production releases will follow on a one year cycle]

2.1.4 Approach and Methodology

The duration of an experiment like ATLAS is one order of magnitude longer than the current pace of change in computing. It is not even clear whether many of the OO technologies currently adopted by LHC experiments for their computing models and prototypes will still be relevant when they start running in 2005.

We don't think it is prudent to assume that the protocols or the actual code used in the early versions of the framework will be in use at experiment start, not to mention through its expected 15-20 years lifetime. Many of the existing frameworks provide satisfactory "horizontal modularity", the ability to replace existing plug-in modules. On the other hand, most of them lack any kind of "vertical modularity", the ability to replace the implementation of the module interface (the "software bus").

Many HENP collaborations have already found themselves locked into using obsolete programming languages or operating systems because of an early implementation choice. In such cases pressure to replace obsolete technologies increases over years which the management resists because they (rightly) consider them to represent dramatic changes, leading to a crisis at the worst possible moment and typically to a complete rewrite of the infrastructure, followed by a painful adaptation of the physics code to it. We believe vertical modularity is the right approach to promote evolution of the software infrastructure to incorporate new technologies as they become useful.

To promote vertical modularity we intend to base module interactions on interfaces described in an external dictionary. An example of such a dictionary is the one on which OMG CORBA is based. CORBA (Common Object Request Broker Architecture) is an industry supported architecture standard that manages communication among distributed components. Distributed objects in CORBA interact using a language neutral interface specified by the programmer using the Interface Definition Language (IDL). CORBA implementations provide an IDL compiler which reads in the dictionary and produces the "glue" code necessary to broker the component interactions.

In a similar fashion our framework will ask all developers of service and physics analysis modules to provide a dictionary file describing their interface to the external world (or to comply with one of the standard interfaces available in the repository). We will use the description to automatically generate code connecting the modules to the framework in use and we will provide tools to assist developers in implementing the interfaces they describe in their target language. If, for example, they will be coding in C++ we will generate a complete header file and, optionally, the skeleton of its implementation to be completed by the developer.

John Milford (LBNL) is developing a set of tools based upon an IDL compiler he has developed using the JavaCC and JJTree tools[8].

This model will allow us for example to replace a framework implemented in C++ with one written in Java, or vice versa, without the need to touch any of the existing modules or to interchange user modules or other components written in different languages without changing the interfaces to those components. Central to our software engineering approach will be the use of open industry standards, of component software connected using a framework, designed in a manner which will allow us to track the evolution of said standards.

The use of an interface dictionary enables us to distribute modules in different threads and processes using standard tools such as CORBA itself. We don't anticipate at this time the need to have distributed physics analysis modules (although this may be the case if we want to support the use a module being developed by a user on a remote client). On the other hand, we do see the advantage of distributing some service modules, such as a remote object server for calibration tables; a remote event display client receiving reconstructed event data from the framework; or a production master process managing a number of reconstruction workers, monitoring their status, load balancing them, etc. (CORBA is already being used in a similar manner in many HENP DAQ and online systems.). The most important service module that we will implement as a separate process is the user interface. The ability to control and configure the framework remotely rather than from inside the application itself will improve the UI (or GUI or Web-based User Interface) interactive response and it will make the framework more robust.

A common objection to the approach of using an interface dictionary is the additional burden it imposes on the developer who has to learn yet another language and use yet more tools. We believe that if the language is an industry standard like IDL, this burden is minimal and in any case more than balanced by the advantages mentioned before. If, as it seems likely, Java will become a popular language among the HENP developers in the next few years, this objection may vanish altogether as Java

---

[8] http://www.suntest.com/JavaCC/

offers with its core libraries very good support for an interface dictionary, for distributed components and even for a multi-language environment.

2.1.5 Management, Organization and Work Schedule

2.1.5.1 LBNL

Craig Tull will be the project manager responsible for organization of the U.S. ATLAS Control Framework software effort. Based on discussions with ATLAS Computing management we anticipate that David Quarrie will be responsible for coordinating and ensuring consistency of the ATLAS Control Framework effort, software design process, data management effort and other ATLAS core software activities. We present later a resource loaded work plan for fiscal year 2000 as well as a list of project milestones through fiscal year 2005. Chris Day will contribute to the control framework development effort and specifically to the software design process (USDP, Unified Software Development Process).

The selection of the control framework as LBNL-ATLAS computing's primary core domain of responsibility has been made in close consultation with the overall ATLAS Computing Coordinator. This consultation is ongoing and should result in an explicit agreement on LBNL's role, responsibilities, and deliverables. The recently concluded "Architecture TaskForce"[9] has set the general plan for this activity while the emerging "Architecture Team" is developing specific deliverables for this and other software efforts. There are three LBNL members (P.Calafiura, D. Quarrie, and C.Tull) on the Architecture Team. This team is expected to provide an overall architectural design for the framework and other systems and to define specific deliverables in terms of components and interfaces. We have based our current plans upon logical assumptions of what the final ATLAS architecture and specific deliverables will be. We expect to adjust our schedule and work-plan to accommodate requests from the ATLAS computing coordinator and the results of the Architecture Team.

We will engage physicists in ATLAS and in other experiments in specification of requirements and in design and implementation discussions. In particular, we will continue discussions with BaBar, CDF, CLEO, D0, and STAR researchers with whom we have already had extensive discussion of these issues.

As well, we expect to draw upon U.S. and non-U.S. ATLAS collaborators in each of our test and documentation phases. Real-world tests of the framework will provide the greatest volume of user feedback on the design and implementation of the system as well as the most accurate and realistic QA and performance measures for the framework.

2.1.5.2 BNL

We propose to draw on our RHIC software experience at BNL to contribute to ATLAS core software development and specifically to the implementation of distributed computing capability and support tools in the offline environment. These tools will be distinct from and layered over the control framework (they are being developed in an experiment-independent project) but closely integrated with it.

The size and geographical distribution of both ATLAS as a whole and U.S. ATLAS make distributed computing support in the framework essential: the ability to use the analysis framework interactively or in batch as a thin local client giving transparent access to data and computing resources at a remote regional center. The rapid growth in network capacities and technologies now underway make a powerful distributed computing capability both possible and practical to a degree not seen in previous generations of HEP experiments.

The STAR BNL group is engaged in an internally funded (LDRD) project developing such distributed computing capability in a form in which it can be applied to ATLAS as well as RHIC and other projects. This project, a Networked Object-based enVironment for Analysis (NOVA), is now underway and has seen first application of its initial components in STAR distributed database tools and ATLAS demonstration prototypes[10].

BNL is responsible for much of the core software development activity of the RHIC experiments. For the STAR experiment, overall computing management and the core software infrastructure development team are located at BNL. The BNL team is

---

[9]http://www.cern.ch/Atlas/GROUPS/SOFTWARE/OO/architecture/

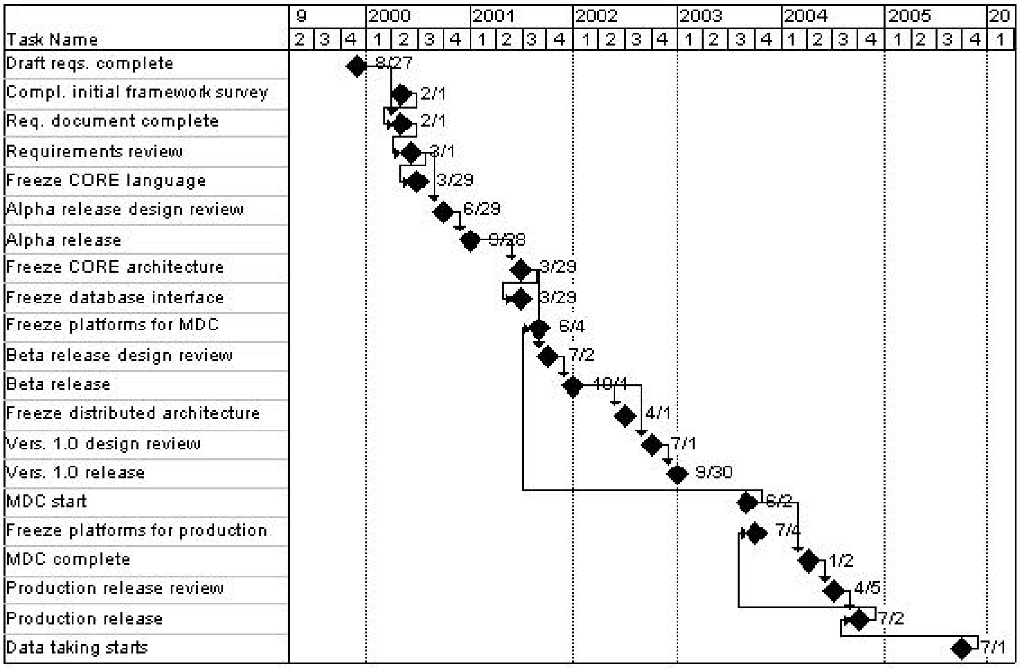
[10] http://duvall.star.bnl.gov/nova

responsible for the development of the STAR analysis framework now successfully deployed, which makes extensive use of the ROOT analysis framework.

Our internal laboratory funding will support a limited transfer of effort in FY00 (~1FTE) of computing professionals currently working on NOVA development and STAR framework development to leveraging this work and experience into an ATLAS control framework contribution. Torre Wenaus will supervise and participate in the work, with further contributions from other BNL physicists. Computing professionals who will participate include Sasha Vanyashin and Victor Perevoztchikov. We anticipate an FY01 request for project support at the 1 FTE level in order to continue this effort the following year.

2.1.6 Control/Framework Milestones

The major milestones for development of the Control/Framework Software up to the initial operations phase in 2006 are given below.



2.1.6.1 Prototype Release (pre-Alpha): May 2000

This release will deliver a prototype reconstruction framework based upon the architecture, delivering the following functionality:

- Support for the existing preliminary ATLAS event and detector descriptor models and event graphics
- Support for multiple Modules merging several detector systems
- Support for data output and input between user modules
- Extensible user module interface

The feature set of the prototype should also include:

- Dynamic loading of user modules
- Sequences with branches

- Rudimentary interactive user interface

This prototype release will provide the first introduction for the ATLAS developer and user community to the overall structure and interfaces of subsequent releases. This prototype will have several distinct advantages over the current provisional system; It will provide a way of working with multiple detector systems; it will provide dynamic loading of modules; it will allow for checkpoint/restart functionality (between modules) of an analysis process. This prototype will also provide a venue and context for discussions of the design, requirements, feature list, and use cases informing the work of the ATLAS Architecture Team and LBNL control framework developers. It will also serve as a working framework and an interface specification for testbeds and development efforts in associated domains and detector subsystems.

2.1.6.2 Test Release #1 (Alpha): September 2000

This release will demonstrate the basic functionality of the package. We expect it to be adequate for use by application code developers but not for casual users. A preliminary version of the application interfaces will exist, but these interfaces will not yet be frozen. The functionality in this release will include: dynamic loading of I/O and analysis modules, run time specification (via a configuration script) of the order of module execution, and the ability to process events through multiple analysis paths with the possibility to discontinue event processing based on the filter decisions signaled by the modules. The release will include an interface to at least one Analysis Tools Package that allows for the creation of histogram/ntuple files but it may not support interactive use of that Analysis Tool from within the Framework. Framework and module configuration information will *not* be permanently recorded to a file or database in this release.

2.1.6.3 Test Release #2 (Beta): October 2001

This release should be adequate for early testing by physicists who are not core participants in application development. We expect all I/O interfaces to be finalized by this time. The additional functionality relative to the Alpha release includes support for recording Framework and Module configuration information to a flat file or database and the ability to reconfigure from that file, an interactive interface to at least one Physics Analysis Tool, support for an abstract interface to additional graphical applications (such as event displays) and support for a set of module and Framework monitoring tools that allow developers to gather statistics on CPU usage, memory usage, etc. This release need not support multithreading nor support tools for distributed configuration across multiple clients. A Graphical User Interface (GUI) is not a milestone for the Beta release.

2.1.6.4 First Production Release: October 2002

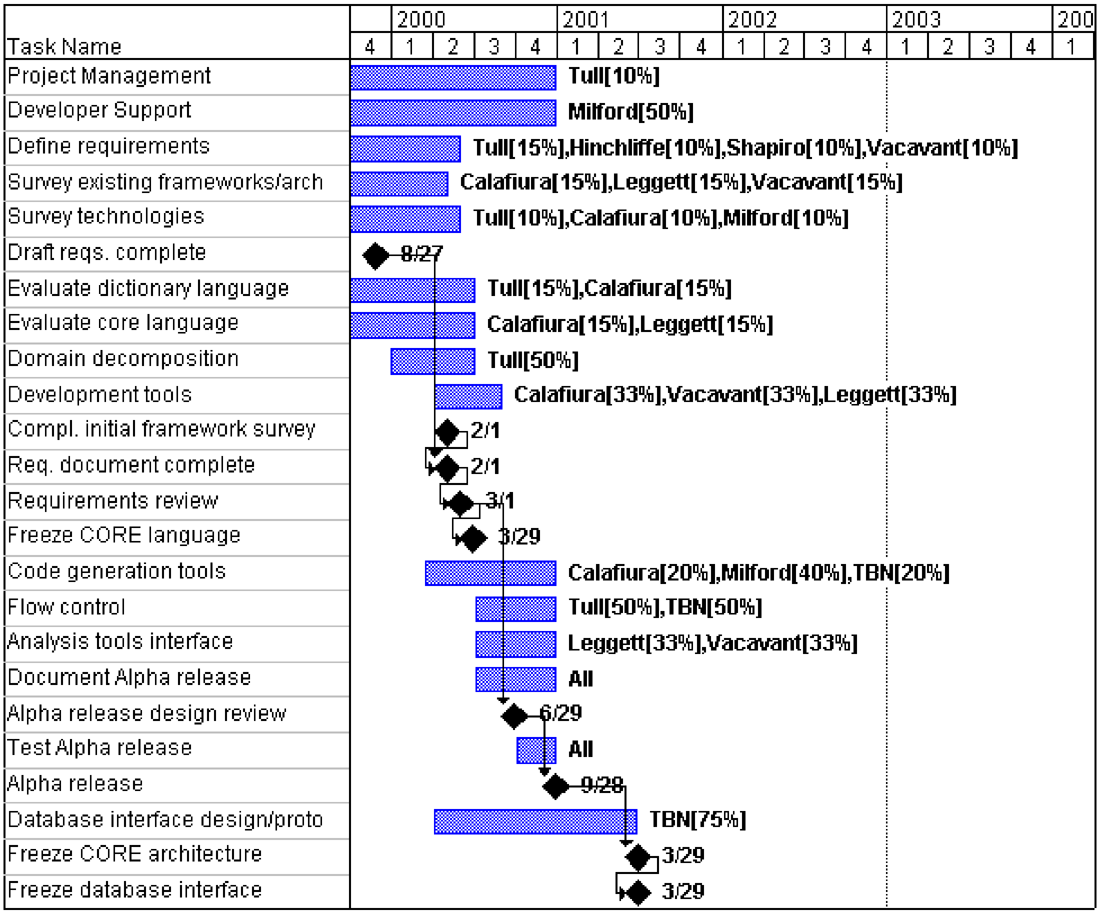
This release is expected to be fully functional and adequate for production operations. It will support multithreading and will include a Production Manager that can be used to control distributed applications. GUI support will be included.

2.1.6.5 Second Production release May 2004

This release will incorporate design and implementation revisions arising out of the ATLAS Mock Data Challenge and other experience with the software. Following this release, we anticipate an iterative cycle of refinement in light of experience and new production releases with a one year cycle.

2.1.7 Control/Framework Plan in FY00

A preliminary resource loaded schedule for development of the Control/Framework software by LBNL personnel in FY00 **only** is given below.



A more detailed description of the tasks and milestones is given below.

Project Management

Project management will consist of coordinating effort within the control framework project and between the control framework project and other software projects within ATLAS.

Developer Support

Software developers in the control framework project will require support for development tool installation and maintenance, maintenance and integration with the ATLAS software development environment, etc. We have also included here the 0.25FTE of support needed for Physics simulation and related items.

Define requirements

We will define a set of requirements for the control framework which do not pre-suppose a particular architecture or solution. These requirements will be defined in cooperation with the ATLAS Architecture Workgroup and with input from ATLAS physicists. A requirements section of the design document for the ATLAS control domain already exists and can form a starting point for this document. These requirements will be the measure of the suitability of the control framework design for ATLAS.

Survey existing frameworks and architectures

There are several examples of existing frameworks for large applications both HENP specific (e.g. AC++, Gaudi, OpenScientist, PAW/Root, StAF) and non HENP specific (especially those based on the various component architectures) that try to address similar issues than the ones concerning us. We will survey them to extract every possible relevant requirements, design patterns and implementation choices.

Survey existing & upcoming technologies

We will survey the three major component architectures (CORBA, DCOM and JavaBeans) and their evolution to determine which of them we should adopt as our (model for a) software bus technology

Evaluate dictionary language candidates

We will evaluate the most convenient meta-data languages to describe the framework interfaces to the external world and to the software bus (e.g. IDL, ODL, SWIG, XMI, proprietary)

Evaluate core language candidates

Although there seems to be no doubt that on a time scale of three-four years most of the algorithmic code of ATLAS will be written in C++, this does not determined a priori which programming language we should adopt for the framework core. We will evaluate possible alternatives (Java) to determine whether they offer enough extra features to make up for the extra complication due to multiple languages issues. Also we will evaluate the impact of a possible future change in the core language of choice and try to deign the framework to minimize it.

Domain Decomposition

We will identify sub-domains of the framework that can be designed and implemented in parallel by separate individuals and/or teams.

Development Tools

We will determine which tools to use for the design, the development, the code maintenance and the documentation.

Code Generation Tools

We will evaluate which area of the framework implementation can be automated by generating code from the framework interface description. We will use existing tools (e.g. SWIG) and/or custom developed ones.

Execution Flow Control

We will develop tools to control the execution flow both via scripting languages (and possibly configuration files) and an interactive user interface

Analysis Tools Interface

We will provide a platform independent interface to multiple histogramming tools (e.g. PAW, ROOT, JAS) allowing the physicist to analyze the data accessible from the framework with her preferred physics analysis tool

Document Alpha Release

The Alpha release will include sufficient documentation for testing evaluation of the design and for testing of the system by non-developers.

Test Alpha Release

The Alpha release will be tested for functionality and for bugs by both software developers and by physicists.

Database interface design & prototype

We will define an interface to the ATLAS Data Model that will allow transparent access to the data whatever their location and storage technology in use will happen to be.

2.2 Data Access and Management Software (WBS 2.2.1.2)

Crucial to the success of the ATLAS experiment will be its ability to deal with enormous data volumes – petabytes of data annually – in a reliable and efficient way. The U.S. has been asked to provide overall database leadership, as well as coordination of database efforts in two of four ATLAS subsystems. The following sections outline how we propose to discharge these responsibilities. With recognized expertise in scalable data handling (cf. the PASS (Petabyte Access and Storage Solutions) project and the High Energy and Nuclear Physics Grand Challenge project), and the combined experience of BaBar, the Fermilab experiments, and the Relativistic Heavy Ion Collider experiments, the U.S. is well positioned to make significant contributions to ATLAS in this area.

2.2.1 Scope of the ATLAS Data Access and Management Domain

The data access domain encompasses both the *infrastructure* needed to manage, organize, store, and retrieve petabytes of data efficiently, and the design, instantiation, deployment, and maintenance of *concrete stores* for physics data.

Infrastructure components include

- control/database interfaces, and generic database components needed to support those interfaces;
- transient/persistent interfaces;
- physical data clustering and storage optimization;
- data organization and indexing for rapid retrieval;
- infrastructure for distributed database development;
- infrastructure for distributed data access by physicists;
- tertiary storage access and management;
- database coding rules and endorsed practices.

Infrastructure work further encompasses assessment, prototyping, and scalability testing of proposed approaches and technologies for data storage and access.

Among the concrete physics data stores are

- the event store, which includes many layers of simulated, raw, and reconstructed data behind a unified event access interface (see the ATLAS Computing Technical Proposal for additional detail);
- detector description stores, which include geometry conditions databases, calibrations and alignments
- run conditions
- trigger databases
- statistics and analysis stores, which support production logging and the user-specific output of individual physicists' analyses.

While fabrication databases (sometimes called production databases) fall under the purview of the ATLAS construction project, interfaces between offline software and fabrication databases must also be defined, developed, and supported (offline analysis, calibration, and diagnostic software may require access to detector construction information).

The distinction between infrastructure and physics data stores is, of course, not a precise one, and there are many essential components that fall squarely on the boundaries between infrastructure and physics content. Event collection management, for example, and more generally, indexing and physical clustering for efficient retrieval, involve both the machinery by which data are organized, requested, and delivered, and event-specific content.

2.2.2 U.S. Participation

2.2.2.1 U.S. Leadership

David Malon of Argonne and R. D. Schaffer of Laboratoire de l'Accelerateur Lineaire are co-leaders of the ATLAS database effort. The U.S. database role therefore entails more than responsibility for implementation and delivery of specific components of the database architecture. It also requires

- oversight and participation in the design of database components for which primary implementation responsibility will lie outside the U.S.;
- participation in the specification and design of non-database components whose database implications are substantial, such as detector description and event model;
- participation in the specification and design of interfaces between offline and fabrication (production) databases.

The fact that the U.S. has responsibilities beyond those associated with specific U.S. deliverables has strong advantages – it can serve to ensure that U.S. database efforts are, from their inception, well integrated into global ATLAS computing plans.

Database coordinators for two of the four ATLAS detector subsystems also come from U.S. institutions. Tom LeCompte of Argonne is the tile calorimeter database coordinator. Steven Goldfarb of the University of Michigan is the muon spectrometer database coordinator.

2.2.2.2 Technical Participation

We propose that the technical contributions of U.S. groups to ATLAS data management efforts be focused in four principal infrastructure areas:

- Infrastructure for distributed database development, and for wide-area access to physics data. Without this infrastructure, serious contributions from outside CERN are difficult, if not impossible.
- Design and implementation of the control/database interface, and of generic database components needed to support that interface. These components are necessary for the deployment of the control framework; the leading role the U.S. will play in the control framework as well as the database domain makes this a task well suited to the U.S. program.
- Database scalability, which is an area of particular ANL strength. Scalability must be addressed in the design stages, even though – insidiously – design failures here may not become apparent until significant amounts of data arrive.
- Definition of the Event Model and interfaces to Control/Framework and persistent data. The Event Model is the representation of event data used in the offline software. U.S. ATLAS includes many of the event model designers for BaBar, CDF, D0 and STAR. This expertise together with strong U.S. involvement in control framework and persistency makes the event model and its interfaces well suited to U.S. involvement.

In support of design and development of concrete physics event stores, we propose U.S. responsibilities in:

- Access to event data from the physics TDR, and to calorimeter testbeam data;
- Data management efforts in support of simulation;
- Detector description, particularly infrastructure to support specification of geometry and detector organization.

2.2.3 Current Efforts

Current work in the database arena is proceeding primarily along the following fronts:

Event data access

This ongoing effort involves making subsystem hit and digitization data from the (FORTRAN/Geant3-based) simulation effort in support of the combined performance technical design report available through the current provisional object-oriented reconstruction and analysis framework known as PASO.

Detector description

While this is not formally a database task, it is a required precursor both to any detector description database and to further (Geant4-based) simulations. The current effort involves providing an XML specification for detector geometry; it is being driven by the database domain, and subsystem-specific work is being done and/or coordinated by the respective subsystem database coordinators.

Infrastructure for distributed database development

For a variety of technical reasons, checking out and building database code is significantly more complicated than checking out and building non-database code. Current efforts involve providing a unified and exportable framework based on the ATLAS Software Release Tool (SRT) for using and developing database code, and support of the PASO framework based on data stored in Objectivity/DB.

Production use of Objectivity/DB

Thanks to a pilot project begun in the late spring of 1999, U.S. database efforts have delivered what is the only current production use of Objectivity/DB in ATLAS. The software, which provides access to tile calorimeter testbeam data, supports an innovative detector-centric view of the data, and serves as a testbed for several transient/persistent mapping strategies simultaneously. Details are available in a recent talk[11], and in the Proceedings of the International Conference on Computing in High Energy and Nuclear Physics (CHEP 2000) (to appear).

Objectivity scalability studies

Simona Rolli (Tufts) has been working on optimization studies based on the use of segmented VArrays in Objectivity/DB. SegVArray is a multilevel, variable-size array with the same interface as the Objectivity/DB VArray class, but containing an ooVArray of SVArraySegments, each of them containing an ooVArray of objects that are elements of the SegVArray. Simona has been working on benchmarking the current ATLAS raw data model, based on ooVArray class, and an extension of it based on the use of SegVArray. The advantages of the SegVArray based approach compared to the ooVArray class based data model may be very large. Simona will present a paper at CHEP 2000 on the performance benchmarks performed with a "toy" data model and with the ATLAS data event model. The Tufts group has had long-time involvement in the ATLAS computing model, ATLAS data model studies, and object databases.

2.2.4 Planned U.S. Database Efforts

2.2.4.1 Database Infrastructure

Database infrastructure will be our earliest focus; without much of this work, further development cannot proceed. High priority will be given to infrastructure for distributed database development and deployment, an immediate need, as well as data access mechanisms (and Objectivity-resident data sources) in support of the control framework prototyping effort led by LBNL. Proposed U.S. tasks include

- Definition of a reference database to contain standard ATLAS schema for use by developers, based on the experience of BaBar and other U.S. experiments;
- Provision of a mechanism by which a reference database may be built remotely, equivalent to building a software package checked out of a repository;
- Establishment of protocols for sharing schema, for avoiding schema conflicts among developers, and for importing schema from outside;
- Devising of a risk-averse strategy for use of the Objectivity/DB database product within ATLAS, and implement it in database coding standards;
- Development of database coding and interaction rules and standards akin to C++ coding rules;
- Establishment of mechanisms and precedents by which large databases may be delivered to outside institutes.

The high priority of infrastructure tasks fundamental to subsequent development are reflected in our milestones; most of the milestones related to this effort appear early in the our software plans.

---

[11] http://www.cern.ch/Atlas/GROUPS/SOFTWARE/INFO/Workshops/9908/slides/thu.3/index.htm

2.2.4.2 Control/Database Interfaces

The interface between control framework and data stores is a highly collaborative area for development, and will rely on interactions not only with control framework developers, but with offline software developers defining the shapes and (persistent) states of data objects – not just events and their properties (such as tracks and hits) but also calibrations and geometries – and their associations to other objects. Early tasks include

- Design and implementation of an approach to maintenance of database and transaction context as control flows from component to component;
- Design and implementation of the means necessary to refer to, locate, and move persistent data into and out of software components that use the control framework;
- Participation in the definition of the approach to transient/persistent mapping as part of the global architecture effort, and provision of the control mechanisms needed to support this approach.

Control/database interface milestones are timed to support and enable delivery of the control framework software, whose milestones are summarized above.

2.2.4.3 Scalability

Database scalability, and more generally, architectural scalability, are properties that cannot be added like one more new feature to software after it has been engineered; scalability must pervade system design from its inception. Much of the proposed effort accordingly involves collaboration with other efforts for which the U.S. will not take primary responsibility, though there are certainly many specific components of approaches to database scalability that we do propose to deliver, as well as component scalability assessment efforts that can be undertaken independently.

It should be understood, then, that a portion of this effort is an ongoing *collaborative* role – we may not take primary responsibility for event collection management, for example, but we *will* participate in ensuring that the design and implementation are scalable to petabytes of data and billions of events. Design scalability efforts must begin early, and persist through the lifetime of the project.

Evaluation and assessment of the scalability of specific technologies and approaches will be accomplished in the following ways:

- By understanding the experience of current and near-term experiments like BaBar, RHIC, and Fermilab Run II;
- Through common projects with experiments that plan to use the same technologies (*e.g.*, jointly with other LHC experiments via RD45 for Objectivity/DB – efforts in which the proposers already have a track record of significant involvement);
- By using the calorimeter test beam analysis project, which will, for example, provide information about the performance of distinct transient and persistent data models;
- Via other short or medium duration projects that will be designed to evaluate approaches to data handling in the context of applications that address genuine physics needs.

These scalability assessments must be accomplished on timescales that match ATLAS decision timetables, e.g., for database selection.

Delivery of specific components for scalable data handling (e.g., prefetching mechanisms for data on tertiary storage and parallel query and analysis capabilities) will be essential to the success of ATLAS computing, but not to the first years of software development.

2.2.4.4 Databases and Event Model

The Event Model is an important ingredient of the ATLAS core software infrastructure and is closely coupled to the data management effort and the control framework. The management of the Event can broadly be broken into its transient and persistent components. The transient services include the memory management of the raw and processed event information and the tools by which clients add, extract and navigate between objects in the Event store. The persistency services must supply mechanisms to efficiently store and access data on external media. The persistency service interface will be independent of the database technology that is adopted, although optimization for each database must be made possible.

The Event Model must define:

- The type of objects and their form that can be stored in the Event.
- The client's (logical) view of the Event.
- The physical organization of the Event, which may differ from the logical view as seen by the client.
- The client's interface to the Event, i.e., tools to allow clients to add and extract objects to the transient store.
- The mechanism to navigate between objects, for example between a given track and the hits that correspond to the track. This is a non-trivial issue closely coupled to the persistent model that is adopted.
- The mechanism to define persistent capable events, that is, the ability to store different levels of detail of the Event or only events that pass the client's requirements.
- Organization of Event collections.
- Mapping of persistent objects to transient counterparts and vice-versa. In an ideal case, the client need not have to define or have knowledge of the persistent shape of an object. While accessing objects from a persistent store, access can be made possible on demand only without the client having any knowledge. There is a close coupling of this point to the choice of the database technology itself as stated before and hence a close interaction between the Database and the Event Domain is a necessity to address and provide a solution for this issue.

The close coupling of the Event Model to data management and control/framework software and the substantial U.S. experience in event model software make it an ideal candidate for U.S. participation. U.S. collaborators include many of the designers and developers of event model software for current experiments: D0 and STAR at BNL; BaBar and CDF at LBNL. The BNL group in particular is interested in combining local expertise and interest in event model software with BNL's role in LAr offline software development to contribute to the development and real-world prototyping of the ATLAS Event Model, with BNL's LAr OO reconstruction development effort serving as a prototyping and testing environment.

BNL participated in the design of the DØ event model (EDM) and the persistence mechanism (DØOM). DØOM is a general layer that decouples the DØ software from any underlying I/O packages, in use for all the I/O of C++ objects, e.g. event I/O, access to calibration databases, and geometry files. It currently has interfaces to serial formats (DSPACK) and relational databases with a CORBA interface in development. The implementation of DØOM was done mostly by Scott Snyder (BNL Physicist) and represents over three man-years of effort. The possibility of extending DØOM to support object databases and employing it in ATLAS is under discussion. In the STAR RHIC experiment BNL played a leading technical and managerial role in the design and development of the event model and its persistent implementation using a specialization of ROOT I/O developed at BNL together with the MySQL relational database. Principal participants at BNL were Valery Fine, Yuri Fisyak, Victor Perevoztchikov and Torre Wenaus.

We propose an Event Model effort closely integrated with the data management and control/framework efforts which is directed at the very early implementation and application of a subset of the event model in the context of a subdetector software project. Our goal is to make very early and direct contact between the core design and development effort on Event Model architecture and interfaces emerging from the control/framework and data management efforts, and real-world application of the Event Model infrastructure. This will allow real-world input at both the design and prototyping stages to the development effort. Specifically, we propose to incorporate into the LAr object-oriented reconstruction development effort being undertaken at BNL the design and implementation of an Event Model supported by the control/framework and data management software emerging from LBNL, ANL and elsewhere and employed by the reconstruction software itself. The milestones for this effort will track the control framework release schedule discussed above, commencing with an initial Event Model implementation operational in the context of the alpha test version of the control framework scheduled for May 2000. This initial prototype will define the reconstruction client's logical view of the event, employ this logical model in application modules and codes, and save processed data in this model using the provided persistency mechanisms.

We present a schedule of activities during FY00, highly coupled to the control framework schedule:

2/00	Event Model schematic design based on LAr reconstruction requirements and design guidelines of the Architecture Task Force Report and early Architecture Team work
3/00	Event Model and application module detailed designs based on interface and module architecture specifications from the data management group and Architecture Team

5/00 LAr OO reconstruction modules based on Event Model operational in the pre-alpha test framework

5-9/00 Revisit design issues based on experience with the prototype

9/00 Alpha version employing features of alpha test framework

FY01 Iterate design and implementation for the full function version of October 2001

The database and event domains have been strongly coupled in the ATLAS software milieu for a number of reasons. The first is that object database vendors have touted the low impedance mismatch associated with the use of object databases; ideally, the transient model and the persistent model can be one and the same. With this approach, defining a physics event model is largely synonymous with determining the shape and structure of stored event data. ATLAS has accepted a reality that falls short of the ideal, with the decision made to separate the transient and persistent representations, but the objective of minimizing the impedance mismatch remains. The second reason is that the daunting scale of the ATLAS event store has tempered the collective thinking – dealing with all those events successfully is, from the point of view of scale, equivalent to solving the database problem.

A survey of Event Model issues illustrates the many boundary overlaps of this domain. The role of the event is an architecture issue, addressed by the Architecture Task Force and the Architecture Team. Access to events by software components is provided by the control framework. The content of an event must be decided by physicists. The shape of an event is an object-oriented design issue for physicists and software developers. The persistent shape of an event is primarily a database issue, informed by the physicists' knowledge of intended use and navigation patterns. The mapping layer between persistent and transient views of an event may or may not be strongly coupled to the database implementation. To the extent that persistent object definitions can be automatically generated from their transient counterparts (as D0 does, for example), this component might be delivered without significant reference to the database implementation. ATLAS has not yet determined what approach to use. To the extent that persistent definitions may be handcoded (for performance or other reasons) or may not be isomorphisms (no one-to-one correspondence between transient and persistent objects), the mapping may require detailed understanding of the database.

With Argonne's leadership in the database arena, LBNL's role in the Architecture Team and control framework, Brookhaven's experience with event models, and the Tile Calorimeter Pilot Project software already in place as a testbed for transient/persistent mapping, the U.S. is very strongly positioned to collaboratively contribute to the ATLAS Event Model. Coupling in also our subdetector software development effort as an Event Model testbed, U.S. ATLAS can contribute to the close coupling between core development and real-world application that is so important to ATLAS.

Reflecting the close coupling of Event Model and Database domains, and recognizing the as yet embryonic nature of our Event Model program, we have organizationally situated our Event Model effort as a subtask of the Data Management effort, with Event Model activities coordinated by Srini Rajagopalan of BNL, who is also the U.S. ATLAS LAr reconstruction software coordinator.

2.2.4.5 Efforts based on existing data stores

Event data from the TDR

U.S. emphasis will be on calorimeter data, in support of Brookhaven's role in object-oriented liquid argon reconstruction and the University of Chicago's role in tile reconstruction (Frank Merritt is the tile reconstruction coordinator). We expect much of the effort to come from the subsystem software projects, with infrastructure to make the data available through the prototype framework coming from core database efforts.

Test beam data

We expect to use the tile calorimeter test beam data as a testbed for production use of Objectivity and for evaluation of transient/persistent mapping strategies. Associated development and extensions of current software capabilities will be undertaken by the tile calorimeter group.

2.2.4.6 Data management efforts in support of simulation

We propose a staged approach to providing data access and storage for Geant4 simulations, roughly in the following order:

1. Monte Carlo events
2. Geant4 hits

3. Geant4 digitizations

4. Detector geometry

Objectivity-based access to Monte Carlo events will provide an input source to Geant4 simulations, will deliver one component of an eventual ATLAS event model, and will serve as a foundation for development of event collection management capabilities. Such an effort is well matched to ongoing U.S. responsibilities (Ian Hinchliffe of LBNL is in charge of Monte Carlo generators for ATLAS), and to work being done at Harvard and Boston University on ISAGEN.

Support for storage and retrieval of Geant4 hits and digitizations must be provided on a timescale corresponding to subsystem use of Geant4 for serious simulation. Delivery of such support will provide the impetus to the ATLAS database domain to address the handling of user-defined object types, and to the overall software effort to further articulate an ATLAS event model.

Definition, prototyping, and implementation of database support for Geant4 hits and digitizations will be undertaken in conjunction with the global subsystem simulation efforts for which the U.S. has responsibility: the inner detector, for which Fred Luehring of Indiana University is responsible, and the liquid argon calorimeter, for which Misha Leltchouk of Columbia University is responsible.

2.2.4.7 Detector description

Providing XML descriptions of detector geometry is an ongoing effort as described above, with much of the coordination and labor coming from subsystem database coordinators. While it is possible to read shape definitions expressed in XML more or less directly into Geant4[12], the work of building a geometry database, with its ensuing advantages and complications, must follow. We propose to continue U.S. involvement in the definition and development of an XML DTD for detector description, and to participate significantly in the specification of a generic model for detector description that must in turn be supported by a database. A focus of this effort will be in the muon subsystem where Steven Goldfarb of U. Michigan is ATLAS muon database coordinator.

2.2.5 Program of work

2.2.5.1 ANL

David Malon will lead the U.S. ATLAS data access and management software effort. This is a natural role for Argonne, given David's Database co-leader role in ATLAS, Argonne's national leadership in Computer Science research, and its pioneering role in introducing object storage and access technologies for petabyte-scale datasets in high energy physics.

David Malon will contribute full time to the project. Two software professionals from Argonne's Decision and Information Sciences Division – John Christiansen and Guy Pandola – will each contribute 50% of their time to this project in FY 00. Guy Pandola's primary focus will be on Objectivity-specific development and John Christiansen will address issues of control/database interface design.

Physicists who will be involved in this effort are Edward May (60%), Tom LeCompte(50%; tile database coordinator), and Robert Wagner (50%), with smaller contributions from Robert Blair, Lawrence Price, and others (summing to approximately 40%).

2.2.5.2 LBNL

David Quarrie, Craig Tull and others at LBNL involved in the ATLAS core effort and specifically in the Architecture Team and control framework efforts will be closely involved on the Event Store work, particularly on overall architecture, control framework interfaces, and application module interfacing and data access. The program will benefit from LBNL involvement in BaBar, CDF and RD45 on event model and related software.

2.2.5.3 BNL

BNL plans to contribute to the ATLAS Event Model as discussed above. How the Event Model effort will be organized in ATLAS is unclear at this time; for now, we include the proposed effort as a Data Management subtask.

---

[12] G4Builder, Stan Bentvelsen

The physicists who will be involved in this effort are: Serban Protopopescu (10%), Scott Snyder (10%), Srini Rajagopalan (50%), Torre Wenaus (20%), with anticipated future contributions from STAR BNL core software members not yet involved in ATLAS. In addition, we propose to hire one software professional who will be able to contribute 100% of their time to this effort. We are able to partially fund this hire for FY00 on laboratory LDRD funds, and we seek FY00 project funding at the 0.5FTE level for the remainder. This person will work with the above mentioned physicists in the framework of the ATLAS core effort, and specifically the database group under the present organization.

2.2.5.4 University of Michigan

The University of Michigan group is a leading participant in the muon subsystem database effort, with Steven Goldfarb serving as muon subsystem database coordinator. We are requesting support for a software professional (new hire) to join the effort and allow the group to broaden their contribution to a more complete framework for developing and testing detector description and associated database software. Because this effort while anchored in muon subsystem work would be applicable across subsystems, it falls under the core database effort. Details of the program are found in the muon subsystem section below.

2.2.6 Milestones

The time evolution of data management activities has several phases. The earliest phase addresses database development and distribution infrastructure, support for data access through the first-year releases of the control framework, and making event data available for reconstruction code development. Somewhat later work supports access to testbeam data through the control framework, with an initial emphasis on calorimeter data, but using the muon testbeam as a testbed for conditions databases by the second year.

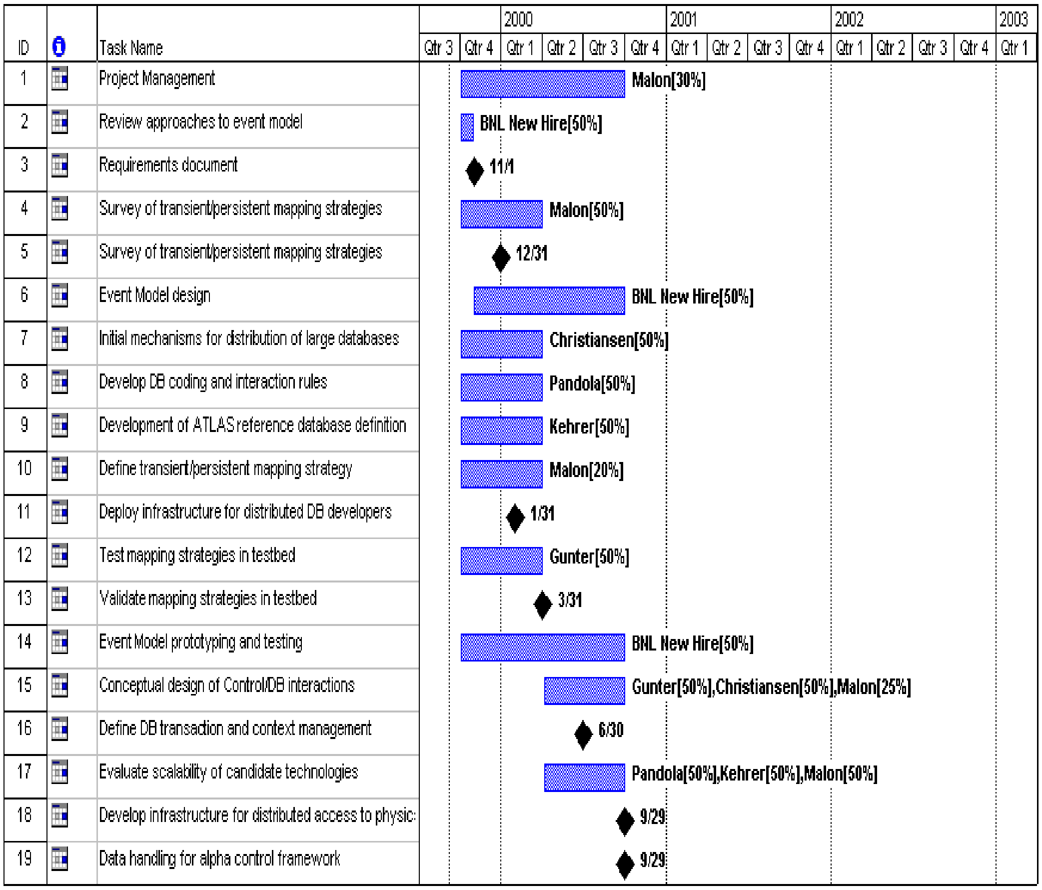
Database support for simulation begins with work on XML specification of detector geometry, and evolves to support Geant4 data access and storage through the control framework by October 2001.

In approximately this timeframe, ATLAS must begin to provide viable incarnations of its various concrete physics data stores – event store, detector description, and conditions databases – and support data storage and access in a production mode.

U.S. database efforts peak in the two years before turn-on. In this phase, we must support reliable storage and delivery of significant data volumes to physics codes. Because petabytes of data begin to arrive at the end of this phase, components that directly address scalability – prefetching from tertiary storage, cache management strategies for concurrent queries, support for parallel database population by reconstruction codes and parallel queries by analysis codes, to name a few – must be implemented in this phase.

Major milestones and resource loaded schedule are shown below. See section 2.4 for a more complete set of milestones.

Date	Milestone
Jan 2000	Survey of transient/persistent mapping strategies completed
Feb 2000	Infrastructure for distributed database developers deployed
April 2000	Validation of selected strategies in pilot testbed completed
July 2000	Database transaction and context management infrastructure defined
Oct 2000	Infrastructure for distributed access to physics data deployed
Oct 2000	Data-handling in support of control framework (alpha) released
Jan 2001	Scalability assessments of candidate data-handling technologies completed
Oct 2001	Release of database infrastructure in support of control framework's full function release
Oct 2002	Release of database infrastructure in support of control framework's production release



Preliminary resource loaded schedule for U.S. ATLAS Database tasks in FY 2000

2.2.7 Tile Analysis Software Testbed

The 1999 Tile Calorimeter Testbeam Pilot Project has delivered an object-oriented implementation of a logical model of the tile calorimeter, and a detector-centric data access architecture. The software provides access to 1998 and 1999 raw testbeam data, including raw runs, raw events, and charge injection calibration data. We have delivered examples that illustrate how to navigate through the logical calorimeter model in C++ to get Tile Element (e.g., towers, cells/longitudinal samplings, PMT ADCs) energies, and how to access raw data (e.g., ADC samples). The architecture supports a simple, object-oriented means for users to provide custom calibration strategies, to associate those strategies with the calorimeter so that energies and timings are computed via those strategies, and to compare calibration strategies by hot-swapping. Example programs illustrating how this is readily done have been provided. A default calibration strategy that reflects the algorithms used in 1998 has also been implemented. The architecture further provides means to reconstruct a run or a user-selected subset thereof using any calibration strategy, and to save the results in an object database. Examples illustrating how to do this have been provided as well. The U.S. has also delivered examples of how to build custom HBOOK ntuples via C++ programs that access data through this model; those examples are also part of the delivered software.

This U.S.-led effort is the first production application of Objectivity to come out of the ATLAS offline software effort.

We have succeeded in FY 1999 in creating an attractive testbed for ATLAS core technologies: an object-oriented framework that provides access to physics data of genuine interest to ATLAS physicists, one that is under our control in terms of ability to experiment with trial implementations of candidate software strategies, including: transient/persistent mapping strategies, common LHC software such as HepODBMS, connection to candidate analysis tools, incorporation of candidate control frameworks, and more. We propose in FY 2000 to use this foundation as the testbed it was designed to be, particularly for technologies and strategies related to core database development. This testbed role requires core database effort in close collaboration with significant calorimeter-specific software development. Calorimeter-specific plans are described in a later section of this document. A particularly attractive option for late FY 2000 will be the extension of this work to the combined TileCal-Liquid Argon test beam run of next summer.

2.3 Core Software Effort in FY2000

The first table below summarizes the FY2000 request for core computing personnel to be supported by U.S. ATLAS Computing, as described in the preceding sections. This table breaks down the FY2000 request represented in Section 3.7.1 of the Project Management Plan. The second table summarizes the planned FY2000 contributions of physicists supported by the base program.

2.3.1 Requested Computing Personnel (supported by U.S. ATLAS) for Core Software

WBS	ANL	LBNL	BNL	U Michigan
2.2.1.1 Control/Framework Software		C. Tull 1.0 D. Quarrie 0.5 C. Day 0.5 J. Milford 0.2		
2.2.1.2 Data Management Software	D. Malon 1.0 G. Pandola 0.5 J. Christiansen 0.5		New Hire 0.5	New Hire 0.5

2.3.2 Physicists supported by base program working on Core software

WBS	ANL	LBNL	BNL	U Michigan
2.2.1.1 Control/Framework Software		I. Hinchliffe 0.5 M. Shapiro 0.2 P. Calafiura 0.6 C. Leggett 0.6	T. Wenaus 0.2	
2.2.1.2 Data Management Software	T. LeCompte 0.5 E. May 0.6 R. Wagner 0.5 R. Blair, L. Price, and others 0.4		S. Rajagopalan 0.5 S. Protopopescu 0.1 S. Snyder 0.1	S. Goldfarb 1.0 (muon subsystem)

2.4 Summary of Core Software Milestones

2.4.1 Control/Framework

May 2000	Prototype release of reconstruction framework
Jun 2000	Alpha release design review
Sep 2000	Alpha release of control framework (basic functionality)
Mar 2001	Freeze beta architecture and database interface
Jul 2001	Full function release design reviews
Oct 2001	Full function release of control framework (general use)
Apr 2002	Freeze distributed architecture
Jul 2002	Control framework V1 design review
Oct 2002	Control framework first production release
Jan 2001	Control framework V2 design review
May 2004	Control framework second production release (post-MDC)
	[Iteration on design reviews and production releases will follow on a one year cycle]

2.4.2 Data management

Milestones marked with an asterisk (*) are timed to coincide with control framework releases.

Data access and management infrastructure milestones

Mar 2000	Infrastructure for distributed database developers deployed
May 2000*	Data handling in support of pre-alpha control framework released
Jul 2000	Database transaction and context management infrastructure defined
Jul 2000	Validation of selected strategies in pilot testbed completed
Sep 2000*	Data handling in support of control framework (alpha) released
Oct 2000	Infrastructure for distributed access to physics data deployed
Feb 2001	Scalability assessments of candidate data-handling technologies completed
Oct 2001*	Release of database infrastructure in support of control framework's full function release
Oct 2002*	Release of database infrastructure in support of control framework's production release

Event data access

May 2000*	Objectivity-based access to physics TDR data
May 2000*	Event model sufficient to support pre-alpha control framework completed

Detector description milestones

Feb 2000	XML DTD finalized after review
Jun 2000	Initial subsystem geometries available in XML
Sep 2000	Detector description generic model--geometry and logical detector organization
Oct 2000	Initial subsystem readout geometries available in XML

Feb 2001	Detector geometry and organization stored in Objectivity

Database support for simulation

Apr 2000	Objectivity-based Monte Carlo event input to GEANT4
May 2000	Initial event collection management (Monte Carlo input collections)
Aug 2000	Objectivity-based storage of GEANT4-generated subdetector hits
Nov 2000	Objectivity-based storage of GEANT4-generated subdetector digitizations
Apr 2001	GEANT4 access to subsystem geometry stored in Objectivity
Oct 2001*	Data handling in support of GEANT4 simulation via control framework's beta release

Database support for testbeam

Nov 2000	Testbeam data access via alpha release of control framework
Jun 2001	Alignment database prototype evaluated muon testbeam

3 Subsystem Specific Software (WBS 2.2.2)

The ATLAS computing group has recently been reorganized, with the new organization reflecting an emphasis within ATLAS on subsystem specific software. There are now four coordinators from each detector subsystem representing simulation, reconstruction, database and offline software. The U.S. (in bold) is appropriately represented in these responsibilities:

	Offline Coordinator	**Reconstruction**	**Simulation**	**Database**
Chair	N. McCubbin	D. Rousseau	A. Dell'Acqua	**D.Malon**/RD Schaffer
Inner Detector	D. Barberis	D. Rousseau	**F. Luehring**	J. Pater
Liquid Argon	J. Collot	J. Schwindling	**M. Leltchouk**	S. Simion
Tile calorimeter	A. Solodkov	**F. Merritt**	A. Solodkov	**T. LeCompte**
Muon	G. Poulard	J.F. Laporte	A. Rimoldi	**S. Goldfarb**
LVL2 trigger		S. Tapprogge		
Trigger/DAQ	S. George		T. Hansl-Kozanecki	H.P. Beck
Event Filter	V. Vercesi	F. Touchard		

The U.S. groups have been active in all areas of subsystem specific software and plan to continue to play a crucial role in developing the software needed to extract physics results from the ATLAS detector in 2005. The details of the software efforts of each subsystem will be explained in the sections below. In general, each subsystem has an interest in developing software in the following areas:

- Detector simulation with Geant4, the new OO simulation toolkit;
- Test beam analysis to verify Geant4;
- Reconstruction software in C++ that runs in the new ATLAS control framework.

In addition, a number of subsystems are active in the database software. Efforts relating to database and control framework software will couple closely to the strong U.S. activity in these core domains, eg. with activities in pixel simulation and LAr reconstruction to serve as testbeds for the evolving control framework.

3.1 Silicon Tracker

The main activity in the silicon tracker software is related to the development of the pixel test beam simulation using Geant4. Other limited activities concern the old legacy software.

3.1.1 Pixel Test Beam Simulation with Geant4

The pixel subdetector simulation, which was performed so far with the Geant3 package, has now to migrate to the Geant4 simulation tool. This transition implies two major activities: the redesign of the software using an object-oriented paradigm and the validation of the algorithmic part (mainly the physics) of Geant4.

Like most of the other subsystems, the pixel group has chosen to start this transition process with the test beam simulation. This choice allows consideration of almost all aspects of this transition while dealing with a small scale project. Furthermore it offers the unique opportunity to test the new simulation tool against real data already accumulated and to come from the pixel test beam facility.

Many tests have already been performed using this test beam setup and more are planned in coming years. The telescope setup includes a set of micro-strip detectors (usually four planes of double-sided detectors) to measure the position, some scintillators and a silicon diode to trigger the data acquisition. The pixel chip or pixel module is placed in a supporting device allowing for spatial translations and rotations. The whole setup is exposed to a high-energy particle beam in the H8 area of the North Hall at CERN and can be placed in a magnetic field.

The test beam simulation project aims to be a testbed for the whole ATLAS pixel system simulation which is to come next year. As such, many parts presently being developed for the test beam simulation will be re-used for the pixel system: the architecture layout, the pixel module geometry, the user-defined material management and physics interaction processes, the appropriate tracking and stepping classes, and the digitization, as well as a wide set of interfaces for utility software (histogramming, visualization, GUI).

3.1.1.1 Status

The project being relatively small and well defined, a pragmatic approach has been adopted, based on frequent iterations to improve both the design and the functionality of the software.

On the geometry side, the third software iteration led to a configuration which is almost the definitive one. It is already as complete as the previous Geant3 description, with a design allowing easy reconfiguration (of the telescope setup for instance).

The implementation and test of the physics interactions and of the tracking inside the material have started. The work is now concentrating on this part.

A new C++ digitization package is also under development. Most of the functionality of the previous one is already available. An interface to the ATLSIM framework allows for intensive checks to be performed within the complete old software chain (simulation + reconstruction).

Finally, a documentation effort is being pursued, with a dedicated web-site for the project[13].

3.1.1.2 Short term plan

Iterations over the code will be continued. The short term issues are the inclusion in the simulation of the pixel module (only single chips were simulated with the Geant3-based test beam simulation) and subsequently the preparation of the first public release of the software, expected for January 2000.

3.1.1.3 U.S. involvement and collaboration

LBNL is in charge of the whole project. Participation of other institutes from Europe is expected for the module simulation and for the comparison with test beam data. Participation from other U.S. institutions is open.

3.1.2 Activities with old legacy software

No real software development activity is taking place in this area. However, there is some participation in maintaining this software.

Some studies are performed using this software and encompass both the simulation and the reconstruction domains in order to provide the various groups (mechanics, electronics) with specific studies since the new OO software is not yet available.For instance, the simulation is being used to check the geometrical acceptance of the pixel end-cap layout, and the reconstruction to study the impact of misalignment of the pixel disks which might be induced by the cooling system.

3.1.3 Reconstruction and visualization:

The UC Santa Cruz group has recently joined the effort to develop the ATLANTIS event display. ATLANTIS is based on the ALEPH event display(DALI). The primarily goal is to develop a tool to check the pattern recognition/reconstruction in the silicon tracker. The work done so far has concentrated on the display of the inner detector tracking information. Obviously ATLANTIS will eventually have many potential uses within ATLAS and these developments are of interest for the whole ATLAS community.

---

[13] http://maupiti.lbl.gov/projects/ptbg4

3.1.4 Future activities

For FY00, the pixel test beam simulation effort will be pursued. It will focus on the comparison of Geant4 to the test beam data and to the Geant3 simulation. Meanwhile the overall design will evolve, easing the transition of the effort towards the simulation of the whole pixel system.

The first goal of this design evolution is to integrate this project in the ATLAS framework which is being developed concurrently: it is intended to take advantage of the LBNL involvement in the latter to transform this simulation project into an application prototype of the framework.

Although the project will initially evolve on its own during the initial phase of the framework development, care will be taken to avoid divergent developments, for instance by adopting in an early stage appropriate application program interfaces. The actual integration with the pre-prototype of the framework will occur before its delivery, expected for May 2000. The integration of the test beam analysis code could be a natural extension of the project.

The on-going work at other institutes in the database/detector description domain will also be considered since it will be an important issue for the whole system simulation. All these activities will help to smoothly ramp up the simulation effort for the whole pixel system during the year. Some coordination with similar Geant4 efforts for SCT will be maintained. The visualization effort will be pursued in the next months. In particular ATLANTIS will be used to compare the existing tracking packages on an event-by-event basis, thus providing an useful feedback to the reconstruction community on how well the existing software does. The future developments encompass areas of more general interest: it is planned to develop an interface allowing to read the existing simulated events and to work on the conversion of the code to OO.

In early FY01, refinements of the Geant4 description of the pixel system will be continued. The comparison with test beam data will in particular be used to ensure a correct simulation of the pixel modules and for the module system tests. At this time, the framework will be in a much more mature stage and a parallel effort should have taken place on the simulation side to guarantee a good integration.

Personnel involved: L. Vacavant, LBL; A. Litke, UCSC.

3.2 TRT

The Transition Radiation Tracker (TRT) is a straw tube tracker combined with a transition-radiation detector for electron identification (electron-pion separation) in the inner detector of ATLAS. Gas-filled straw tubes provide radiation hard tracking capabilities over large volumes at modest costs compared to other tracking techniques. The TRT consists of a barrel part containing over 52,000 axial straws of about 150 cm total length, and two end-cap parts containing approximately 320,000 radial straws[14]. There are over 105,000 readout channels in the barrel TRT alone which must be instrumented.

U.S. ATLAS has responsibility for the construction of the barrel TRT (mechanical construction and electronics integration) and design and production of the electronic front ends for both the barrel and the end cap systems. This responsibility is shared between four U.S. institutions: Duke University, Hampton University, Indiana University, and the University of Pennsylvania. The U.S. institutions have played major roles in software development activities for the TRT. The U.S. has had overall responsibility for simulations for ALL of the TRT. This has primarily been done by Fred Luehring at Indiana University who is the TRT software Coordinator, Inner Detector Simulations Coordinator and the TRT Database Contact. There has been additional work by several other members of the collaboration. Most of the development this far has been aimed at device simulations, testbeam frameworks, detector calibration, and physics studies necessary for the Inner Detector and Physics Technical Design Reports. An overview of this work is presented below.

3.2.1 TRT Simulation

Indiana University (F. Luehring) has overall responsibility for the simulations within the TRT community. The simulations completed up to now have been performed using Geant3. Since the TRT is located in the inner detector (ID) of ATLAS, the materials budget for this subsystem has been a major concern. Simulations have been performed which indicate that the

[14] Inner Detector Technical Design Report, CERN/LHC/97-17 (1997)

materials budget of the TRT would not degrade other subsystem performance beyond reasonable levels, and at the same time it provides robust, continuous tracking and particle identification at large radii in the ID. The TRT provides up to 36 measurement points per track. It consists of mostly low-Z materials, with a total thickness of the active detector of approximately 10% of a radiation length at any pseudorapidity (The figure will be higher in the barrel service flange region). About one-half of this is due to the transition radiation fibers and foils. Detailed simulations of this performance and materials budget were a major part of the ID Technical Design Report (TDR).

In addition to the materials budget, Geant simulations of reconstructed tracks in the TRT, including pile-up expected at the lower luminosity 10^{33} cm^{-2} s^{-1} have been performed for the ID TDR.
The comprehensive study included detailed information about expected rates, occupancies, signal shapes and shadowing effects. Also studied were straw tube resolution and hit finding efficiency as a function of rate F. Luehring completed this work (and continues this activity) and wrote the entire section of the TRT portion of the ID TDR simulations (except the TR performance section).

3.2.2 TRT Test Beam Software

The ATLAS testbeam software development and maintenance has been performed mainly by the University of Pennsylvania (P. T. Keener) and Indiana University (F. Luehring). This includes overall control system (data acquisition, event display and reconstruction) for the TRT test beam runs at CERN. The University of Pennsylvania group has sole responsibility for the testbeam data acquisition system for the TRT. This group shares the analysis software responsibilities with our European colleagues.

There has been extensive comparison of the GEANT simulation with the TRT testbeam data by A. Manara of Indiana University. The test beam data comes from both barrel and end-cap TRT modules test results which has been accumulated over the past several years at CERN. The simulations include test beam setup, event reconstruction and particle identification and analysis (electron-pion separation).

Personnel involved: Duke (V. Vassilis, S. Oh, C. Wang, and W. Ebenstein), Indiana (F. Luehring, H. Ogren, R. Gardner), Hampton (O.K. Baker, K. McFarlane, K. Assamagan), and UPenn (P. Keener, R. Van Berg, W. Williams).

3.2.3 Physics Simulations

Hampton University has contributed to the study of the SUSY Higgs discovery potential using ATLAS. The work has was accomplished using ATLFAST to study the charged Higgs boson decay below the top quark mass. This work has resulted in several talks and two ATLAS notes. The work appears in the Physics TDR[15]. Also for the Physics TDR, Luehring updated a number of ID TDR studies. The TRT charged particle hit rate studies were updated and additional studies of fake tracks and track finding efficiency at full luminosity were undertaken. The results of the fake rate/efficiency studies led to a substantial optimization of the TRT readout. Also the TR performance studies of the ID TDR were updated to include the current geometry, signal shapes, and test beam study TR model tunings. The TR performance update was done without using pileup.

Personnel involved: Hampton University (K. Assamagan), Indiana (F. Luehring).

3.2.4 Future Work

Future work involves making the transition from FORTRAN-based Geant3 to the OO-based Geant4 simulations, as well as OO database management within the TRT. This work is being coordinated with software activity at CERN and elsewhere (D. Barberis; Genova, Italy). At Indiana University (F. Luehring), the first task will be to develop an OO design of the barrel TRT and beginning its implementation with GEANT4. The intent is to create a testbed for the TRT testbeam reconstruction software environment and data, following the lead of the Tile Calorimeter Pilot Project. Much of this work will involve certification of the detector response of GEANT4. For this purpose, a detailed comparison will be made of GEANT3 and

---

[15] Physics Technical Design Report, CERN (1999)

GEANT4 derived tasks, including signal shapes, hit occupancies, and track parameter resolutions. Indiana University personnel additionally intend to participate in the port, certification, and maintenance of TRT code on Linux platforms, using a 32-node Linux cluster at Indiana University for the TRT simulations. The cluster has an AFS client running so that code stored in repositories at the BNL cell are visible. A lightweight client-server Java framework has been written to generate, reconstruct, and store Monte Carlo data on a massive storage system at Indiana (HPSS). The Indiana University personnel involved with the software work are F. Luehring, R. Gardner, A. Kryemadhi, K. Kallbach-Rose. Personnel involved with Linux cluster processing framework are R. Gardner, A.
Kryemadhi, D. Savintsev, all of Indiana University.

Hampton University personnel (K. Assamagan) and some collaborators at Duke University (V. Vassilis) are planning to contribute to the work to make the transition from simulations with GEANT3 to GEANT4. The plan is contribute to comparing the test beam data to the GEANT4 simulations. Additionally, there is a plan (K. Assamagan) to simulate the data from the X-ray setup at Hampton University using GEANT4.

3.2.4.1 TRT OO Particle Identification

A. Manara and H. Ogren have been developing particle identification algorithms based on the use of neural networks. The goal is to improve electron-pion separation, and additionally supply some kaon-pion discrimination. The plan is to convert the existing code to C++ and to contribute to the TRT reconstruction event shape.

3.2.4.2 TRT Data Event Model and Database Infrastructure Work

Indiana University collaborators plan to be intimately involved in the design of the TRT event shape for ATLAS. This will involve participating in the database coding and interaction rules for the TRT data objects; specifying the type of objects making up a TRT event, coding of client interfaces, navigation and persistence selection mechanisms for the event. The goal is for a complete convergence of the entire ATLAS experiment on the use of XML as a data description language.

There is additionally the goal at Indiana to be involved in the effort to define transient and persistent mappings of the TRT data within the context of global data grid efforts. Indiana University has considerable expertise in massive data storage systems configuration, management, and programming. This expertise will assist them in the development of data transfer mechanisms between the Indiana massive storage systems and computational clusters.

For this purpose, we intend to make simulated TRT events accessible as part of the overall distributed framework within the ATLAS software environment. A data cataloging/serving system currently under development for the Fermilab FOCUS Collaboration can be adopted for interim use.

Personnel involved: Indiana University: F. Luehring, R. Gardner, A. Kryemadhi, K. Kallbach-Rose. R. Gardner,
A. Kryemadhi (Indiana High Energy Physics); P. Berg, R. Indurkar, S. Kostov, and L. Lemons (Indiana University Department of Computer Science).

3.3 Liquid Argon

3.3.1 Simulation

The U.S. has played significant role in the LAr simulation in Geant3 and the study of the calorimeter response. This has primarily been done at Nevis, Brookhaven and Arizona. The results of these studies have been documented in the recently published Physics Technical Design Report. Some of the contributions by U.S. physicists in the LAr Simulation in Geant3 and which have been published in the Physics Technical Design Report are :

- Simulation of the narrow strips in the first sampling and the optimization of strip width based on pi0 rejection and pointing studies.
- Optimization of accordion shape for minimal phi-modulation.
- Determination of the optimal depth and granularity of each of three samplings for different lead thicknesses.

- Simulation of the dead material in front of calorimeter and optimization of cryostat wall shape.

The new simulation activities are now focussed toward the usage of object oriented methodology. The Geant4 toolkit allows us to do just that.

In the new formation of the Liquid Argon software organization, Mikhail Leltchouk from Nevis Laboratories has been named the coordinator of the ATLAS LAr simulation effort. He is therefore responsible for coordinating and integrating the LAr simulation effort worldwide. This, together with extensive responsibilities in the LAr construction effort, makes it natural for the U.S. to contribute extensively to the LAr simulation effort. The U.S. is expected to be a major contributor to the simulation of the LAr electromagnetic calorimeters, cryostats and coil (at Nevis and BNL) and the Forward Calorimeter (at Arizona). This includes work in detector description, simulating the detector geometry, digitization, extensive tests to validate the geometry, and study of the response of the calorimeter in different environments. Current work has already started in understanding how to simulate the accordion structure in Geant4.

Current Personnel involved are: J. Dodd, M. Leltchouk, B. Seligman, M. Seman (Nevis); P. Nevski, I. Stumer, F. Lanni (BNL); P. Loch, M. Shupe (Arizona).

3.3.2 Reconstruction

Much of the activity to-date in the LAr Reconstruction software has been done using Fortran and has been the baseline for all physics and detector response studies. These activities concluded mid 1999 with the publication of the Physics technical design report. The effort is now focussed in delivering a complete well-tested object oriented version of the LAr reconstruction software in a time frame of two years. The U.S. groups expressed strong interest in participating the design and development of an OO LAr reconstruction software and have taken a lead role in establishing early prototype components.

During the last six months, the LAr reconstruction group has developed the use-cases for the software, documented existing software, designed and implemented the first working version of the reconstruction software. The design and implementation work has been performed at BNL and Arizona. The software works within the framework of PASO (a Provisional Analysis Skeleton for Object oriented software, provided by ATLAS), implements the infrastructure for the reconstruction, reads Calorimeter digit information from Geant3 data, implements basic cell and cluster finding algorithms and writes out histograms of relevant parameters. This work was discussed extensively at the software meetings at CERN. The second version of the software is expected to be available by mid-February which will implement many of the corrections, geometry constants and provide detailed comparisons with the Fortran based software. We are also exploring the possibility of combining this effort with the Tile Reconstruction algorithms to provide a unified software package. Down the road, the evolution of the LAr reconstruction software to handle calibration, test beam data and other special cases will be addressed.

While much of the reconstruction work is dependent on the architectural issues, we find that early involvement in the establishment of such prototype components can lead to a better understanding of the needs of the architecture. In addition, a strong involvement in the reconstruction issues can provide a test-bed for architectural work, where the U.S. will clearly play a major role. This effort will be realized in the next few months as the architectural implementation gets underway.

Personnel involved: H. Ma, S. Rajagopalan (BNL), P. Loch (Arizona), J. Parsons, S. Boettecher (Nevis), B. Cleland, J. McDonald (Pittsburgh)

3.3.3 Database Activity

The significant Reconstruction and Simulation activity involves some participation in establishing the interfaces and providing access to external information from user software. The ability to read and write Event information into databases should also be possible. The person responsible for establishing the Detector Description parameters required for simulation and other database related interfaces for the Liquid Argon subsystem is Stefan Simion from CERN. We are exploring the possibility of further work with Stefan which will allow a tight integration of the interfaces between these closely related efforts, and exploitation of the significant areas of overlap.

With the recent success of the Tile Calorimeter Pilot Project at ANL, dealing with the ability of storing test beam data into Objectivity and retrieving it for subsequent analysis, the U.S. LAr software group is looking into the possibility of launching a similar project with the help and experience of the Tile personnel.

Personnel involved are M. Leltchouk (Nevis); S. Rajagopalan (BNL)

3.3.4 Test Beam Activity

BNL, Nevis, Arizona, Pittsburgh, and SMU are involved in the current test beam activity. Software activities include the development of test beam algorithms and analysis to understand the response of the calorimeter. The test beam provides the right environment for the study of the detector response and systematics for establishing corrections for offline reconstruction. The possibility of using a common offline framework for analysis of test beam, simulation and real data is currently being explored. We are also investigating the possibility to coordinate efforts with the Tile calorimeter to launch a project to read and write data from an OO framework into Objectivity. Personnel involved includes nearly everyone from all the above institutions.

3.3.5 Calibration

Pittsburgh, BNL and Nevis have taken lead roles in the past in the development of optimal filtering algorithms. We are planning to be involved in the development and study of online calibration procedures and optimization of these algorithms. This work needs to be closely coupled with the offline work and test beam. While the test beam is the natural place for testing all calibration algorithms that are developed, the evaluation and verification of online algorithms should be handled by offline software frameworks. The long term goal is to become involved in the offline physics calibration issues for the analysis and realization of physics in ATLAS.

Personnel involved are J. Parsons, S. Boettcher, M. Leltchouk (Nevis); B. Cleland, J. McDonald (Pittsburgh); P. Loch, J. Rutherfoord (Arizona); F. Lanni, D. Lissuaer, H. Takai, S. Rajagopalan, H. Ma, I Stumer (BNL).

3.3.6 Detector Response and Physics studies

The U.S. groups have contributed significantly to the study of the response of the Liquid Argon subsystem and its impact on physics signatures. These studies have been published in the recently published Physics Technical Design Report. An example of one such study is the ability of the Liquid Argon calorimeter to reconstruct clusters that do not point back to the vertex. Such a possibility exists in some super-symmetric models where a photon is created from a decay of a long lived super-symmetric particle and hence would not point back to the interaction vertex. Special clustering algorithms were written by physicists from Nevis and BNL in order to study such signatures. The U.S. will continue to play a major role in studying such issues and the impact that they will have in our ability to extract physics.

3.4 Tile Calorimeter

3.4.1 Tile Calorimeter Pilot Project

The primary software effort of the Tilecal group over the last 9 months has been the Tilecal Pilot Project. This was the creation of a new test-beam analysis system, using C++, OO design, and an Objectivity data base for the offline analysis of test beam data. The purpose of the project was to gain practical experience with C++ and Objectivity, and to allow further software development to be done in a framework compatible with the final ATLAS analysis system. At present, all the initial objectives of the Pilot Project have been accomplished, and the new C++ code has all the functionality of the old "Tilemon" system Future developments of the Pilot Project will proceed along several lines:

1) Optimal filtering (see below)
2) Further improvements in structure of code and classes.
3) Improvements in existing online documentation
4) Access to muon wall and beam data via the transient model.
5) Added functionality (e.g., using LHC++).

Personnel involved: D. Malon, E. May, T. LeCompte, B. Wagner (ANL).

3.4.2 PASO

The "Provisional Analysis Skeleton for Object-oriented" ATLAS software development is the current temporary framework for developing reconstruction code. Several members of the Tilecal group attended a tutorial workshop on this in November, and we are now developing a transient data record for the "full ATLAS" Tilecal system.

Our hope is to be able to read Geant3 tapes by the February 2000 software workshop, to have a preliminary definition of the major Tilecal classes, and to have begun the reconstruction of Tilecal clusters from the Geant3 tapes.

We have had preliminary discussions with the U.S.-LAr group about common structures for the "cell" and "cluster" classes, and both groups agree this is an important goal. We will be meeting with LAr software people at Brookhaven in January, and will develop more detailed plans then. At present, it seems highly desirable to combine forces for a least a portion of the reconstruction effort, and to have a common framework for reconstruction to the fullest extent possible.

3.4.3 Database/detector description

Tom LeCompte, as database coordinator for the Tile Calorimeter, is currently developing the representation of the Tilecal module in XML. This is the detector description which is the bridge between the database and a generic object oriented model of the detector. This generic model is then used to build the Geant4 geometry description.

3.4.4 Optimal Filtering work for full-ATLAS reconstruction as well as for test beam work

"Optimal filtering" is the process of extracting the best energy deposition for the beam crossing which produces a trigger, by using the measured energy for this crossing as well as the energies of the preceding 3 crossings and the following 3 crossings. Richard Teuscher (Chicago) is developing and testing a method for this, and it appears that we are close to having an optimal precedure for Tilecal in the high-intensity ATLAS environment.

The Tilecal test-beam analysis provides a useful benchmark for this work. Although the testbeam does not have a comparable spill structure, and the time-slices which are read out are not synchronized with the particle interaction, nevertheless this is useful both for studying improvements in Tilecal energy resolution and for measuring the precise time-shape of the Tilecals signal under field conditions. Early work on this was done by Andy Hocker, and is being continued by Ambreesh Gupta (both from Chicago).

3.5 Muon Spectrometer

3.5.1 Simulation and Reconstruction with the FORTRAN software

The U.S. muon community has been active for a long time in developing and using the current muon simulation and reconstruction software to answer detector design and physics questions. We have written many ATLAS Muon and Physics notes and contributed to the Muon Technical Design Report, the First Level Trigger TDR and the Physics Technical Design report. This work also includes simulation and reconstruction of the Cathode Strip Chambers(CSC). Although we will still use this software, we are not proposing to develop it further.

Personnel involved: J. Shank, Boston University; A. Caram, J. Huth, C. Slowe, Harvard; F. Taylor, MIT; K. Sliwa, Tufts; B. Zhou, University of Michigan; P. Nevski, V. Tcherniatine, BNL.

3.5.2 Cathode Strip Chamber simulation in OO

Since U.S. ATLAS has full responsibility for the Cathode Strip Chamber subsystem in the muon spectrometer, it is natural that the U.S. muon community plays the major role in CSC software development. We plan to continue CSC software simulation development in Geant 4 to provide a smooth transition to OO software technologies. We will continue studies of CSC performance and will compare new results with previous simulations in Geant3 that we have developed earlier.

Personnel involved: P. Nevski, V. Tcherniatine, A. Vaniachine, BNL.

3.5.3 Level 2 muon trigger simulation

The Muon level 2 trigger studies have been ongoing for many years now and involve collaboration between groups at Boston University, Harvard, MIT and University of Michigan. The results of the Boston muon level 2 trigger simulation package have been reported in ATLAS-DAQ-99-003. Boston University and Harvard are currently working on integrating this FORTRAN stand-alone program into the official ATLAS software, atrig, using the ATLAS code management system (CVS/SRT). Harvard undergraduate student, Chris Slowe, is working on a C^{++} implementation of the algorithm, which will be included in the Trigger Reference Software. The code is in atrig now and we are currently subjecting it to extensive tests, leading up to production running for the Trigger Technical Proposal which will be published in spring 2000. This production running will use BU's Origin 2000 and the BNL ATLAS Regional Center Linux boxes. The main study we will do is the overall rejection of level 2 over level 1. So far, no study has been done on the correlations of level 1 and level 2. This is a potential problem area because low P_t tracks, which mistakenly pass the level 1 trigger, may also pass the level 2 trigger, especially in regions where the integral B field is small. We will also study the efficiency of the level 2 for specific high Pt muon physics channels. We have been working closely with ATLAS trigger community: Traudl Hansl-Kozanecki (Saclay) and more recently with Stephan Tapprogge (CERN) on the atrig and Reference Software work, as well as Aleandro Nisati and the Rome trigger group.

3.5.4 The AMBER reconstruction package

We plan to further develop the muon track reconstruction package, AMBER. This is a C^{++} package designed using modern object oriented techniques and reviewed by the ATLAS community. It currently needs porting to Linux and algorithm improvements so that its performance matches that of the old FORTRAN muon reconstruction package. It will also need modifications to fit into the new control framework software being developed by ATLAS and interfaces to the detector description and Geant4. In Addition, we are actively testing and helping develop the combined muon reconstruction program, MUID. This package performs a combined fit using data from the Inner Detector, Calorimeter and the Muon Spectrometer to produce the best possible muon momentum reconstruction. Versions of this program were used in producing results for the Physics TDR and devlopment of MUID continues.

Personnel involved: J. Shank, Boston University; B. Zhou, D. Levin, S. McKee, University of Michigan; ; K. Sliwa Tufts

3.5.5 Track reconstruction optimization.

A major task facing the entire muon community is to re-write the muon reconstruction code to make it run significantly faster than its current version. At present, track reconstruction takes significantly longer than required for the Event Filter. The long muon reconstruction time is due to the very complicated magnetic field in the ATLAS detector. Standard technique of separating the pattern recognition and the momentum fitting is not applicable as there is no such thing as a non-bend plane in the ATLAS muon detector. Sorting multiple hit pattern possibilities must involve evolving the track candidates in the magnetic filed, which is CPU intensive. There are great advantages to having the same code used in both the Event Filter and the offline reconstruction tasks, provided that the code is made to run as fast as required by the event rate into the Event Filter. The U.S. Muon group will devote an increasing amount of effort to the task of creating a new, fast, version of the muon reconstruction program. The Tufts computer expert working on MONARC is expected to devote an increasing amount of time to the muon reconstruction problem by the end of the year, when the work on MONARC-type simulations for ATLAS and U.S. ATLAS will be winding down.

Personnel involved: B. Zhou, D. Levin, S. McKee, University of Michigan; K. Sliwa Tufts.

3.5.6 CSC reconstruction software

The CSC subsystem is unique since it has to operate in the highest background rates. Difficult background conditions provide challenges that have to be resolved by sophisticated track reconstruction software. To answer detector design and physics questions and to facilitate a transition to OO software we plan to continue to develop the reconstruction software for the CSC subsystem of the muon spectrometer.

Personnel involved: P. Nevski, A. Vaniachine, BNL.

3.5.7 Software for muon test beam activities

We plan to continue development and support of the software required for the on-line test beam monitoring and analysis at the X5 high-rate facility at CERN. This software is needed to study the performance of new Cathode Strip Chamber prototypes and their electronics under high background rate conditions.

Personnel involved: P. Nevski, A. Vaniachine, BNL.

3.5.8 MDT auto-calibration software

The UM group will develop the muon MDT auto-calibration software package for ATLAS. We will develop the required interfaces, and test the entire chain of the muon simulation and reconstruction based on Geant4 We have developed the auto-calibration code and tested it for barrel muon MDT geometry, and will carry out more studies for the end-cap MDT chambers. We have the Geant4-based muon system simulation software up and running.

Personnel involved: B. Zhou, D. Levin, S. McKee, University of Michigan

3.5.9 Muon Database Software Activities

3.5.9.1 Responsibilities

Steven Goldfarb is currently serving as task leader for the ATLAS muon database system. In this position, he is responsible for the development and maintenance of the event model and detector description database for the muon system, including the software necessary for accessing and storing the data, as well as mechanisms linking the data stores with conditions data (calibration and alignment), and its integration within the overall ATLAS framework.

This work includes the coordination of database software development for the various muon subsystems (MDT, RPC, CSC, TGC, services), as well the organization and construction of a common framework and utilities for the complete muon system. More specifically, it involves the development of mechanisms capable of extracting and translating data between the persistent data store and the generic transient model, and the provision of interfaces between these objects and the application software, including simulation, reconstruction and analysis.

A description of the muon database task is available[16].

3.5.9.2 Accomplishments to date

Steven's initial work concentrated on the development of the muon detector description[17]. This included:

- development of a hierarchical geometry model for the RPC subsystem based on an existing model for the MDT subsystem;
- construction of the associated positional transformation and logical identifier mechanisms;
- implementation of software to extract simulated Geant3 digits for the MDT and RPC subsystems from the existing data stores into the current event model;
- development of mechanisms for the verification of the detector description positional transformations for the simulated digits.

More recently, Steven has worked closely within the ATLAS database group in developing the AGDD (ATLAS Generic Detector Description) model based on XML (eXtensible Markup Language)[18]. This work has included:

- consultation with the development team during the construction of the model and co-authoring of the most recent DTD (Data Task Definition) file[19];

---

[16] http://home.cern.ch/muondoc/software/Database/TaskDefinition.ps

[17] http://home.cern.ch/muondoc/software/Database/Meetings/Status-19990413

[18] http://www.cern.ch/Atlas/GROUPS/DATABASE/detector_description

[19] /afs/cern.ch/atlas/offline/DetectorDescription/AGDD/data/AGDD_2.01.dtd

- development of example barrel muon station geometries in XML and the subsequent presentation of these examples in a tutorial to the muon detector community[20];
- construction of Geant4 geometries from the XML examples, demonstrating completion of the full chain from the XML data through the generic model to Geant4 volumes;
- motivation of commitments from each of the major muon subsystems for the development of the detector description.

The University of Michigan, in particular, has committed to providing the complete description of the MDT subsystem geometry in XML.

3.5.9.3 Work in progress

Current muon database efforts are focused on the following immediate tasks:

- translation of the existing AMDB (ATLAS Muon Data Base) detector description into XML for each of the muon subsystems;
- porting of the existing testing mechanism for the MDT and RPC event model and detector descriptions to the PASO (Provisional Analysis Skeleton for OO-ATLAS) framework[21];
- construction of a muon-specific interface between the generic model and the Geant4 simulation software.

An evolutionary development plan has been mapped out and presented to the muon detector community[22].

3.5.9.4 Future development

The current stage of development for the XML detector description calls for the exact translation of the existing AMDB description, followed by a period of testing and evaluation to ensure the replication of previous results before the elimination of AMDB. Continued development of the subsystem descriptions will require active involvement by the detector experts. This will be an ongoing responsibility for each of the subsystems, but will require significant coordination on the part of the task leader.

The presence of additional manpower to aid in this domain would allow the task leader to provide a more complete framework for development and testing. Specific tasks to be carried out could include:

- construction of automatic mechanisms for extracting geometrical data from the muon layout database to the XML description;
- development of utilities for testing the XML description for internal consistency, consistency between the subsystem geometries, and consistency with the layout data;
- provision of a more extensive testing mechanism for the positional transformations of the generic model.

Development of a preliminary interface between the Geant4 simulation and the generic model of the detector description is planned for spring, 2000. Geometries constructed through this interface will be tested against those currently obtained using a direct extraction of the AMDB parameters. Completion of the remainder of the simulation interface will begin shortly after the evaluation period. In addition, the construction of interfaces connecting the muon reconstruction software to both the detector description and the event model will be started later in the year. This will include the development of mechanisms for the extraction and storage of event data, such as simulated digits, hits and reconstructed objects. Usage of test beam data to evaluate the storage and access of data for analysis is previewed for the following year.

In this case, the presence of additional manpower would be greatly beneficial for the development and testing of the simulation and reconstruction interfaces. Specific tasks could include:

- addition of functionality to the Geant4 interface to facilitate tuning between (processor-intensive) Geant4 volume parameterizations and (memory-intensive) volume instantiations;

---

[20] http://home.cern.ch/muondoc/software/Database/Meetings/MuonTutorial-19991116

[21] http://www.cern.ch/Atlas/GROUPS/SOFTWARE/OO/applications/Paso

[22] http://home.cern.ch/muondoc/software/Database/Meetings/Status-19990914

- investigation of mechanisms to provide partial or complete geometries "on-demand" for muon reconstruction and tuning of the level of detail provided by the description;
- extensive testing of the data storage and extraction mechanisms using muon test beam data stored in Objectivity/DB data stores.

3.5.9.5 Long term plans

In the coming four years, the muon database task must provide a complete and working system, including the data access and storage mechanisms mentioned above, as well as a number of well-tested interfaces to the application software. The final product must also include software for the access and storage of the conditions data, as well as mechanisms to link this data (typically time-dependent calibration and alignment parameters) to the event and detector description.

While a clear plan exists for the construction of the framework software and the coordination of the subsystem contributions, the presence of additional manpower during the development would ensure the presence of extensive testing and evaluation of the database software. Such testing is vital when one considers that much of the technology to be used is new to our field. Furthermore, usage of this new technology may provide unforeseen possibilities which warrant investigation. For example, while the linkage of conditions data to the detector description is foreseen, one may wish to provide similar mechanisms to access pertinent detector data currently being stored in the subsystem facbrication production database. Close coordination between the two systems from the outset, including testing and evaluation, would certainly be greatly advantageous for stress testing the U.S. ATLAS computing effort.

Personnel involved: S. Goldfarb, B. Zhou, D. Levin, S. McKee, University of Michigan

3.6 Background studies

Though not exclusively subsystem specific, the radiation levels in the ATLAS hall have a profound effect on each subsystem. Studies of this background have been ongoing in the US for a number of years. Since March 1998, the University of Arizona has been almost the sole source of ATLAS radiation background studies associated with engineering tradeoff analyses. It was in that month that Michael Shupe was appointed Convenor of the Radiation Backgrounds Working Group, following the departure of Alfredo Ferrari (Milan) to another experiment. The University of Arizona group uses the Geant3 interface to GCALOR, written by Christian Zeitnitz (University of Mainz) while a postdoc at Arizona. Ferrari uses standalone FLUKA, which he maintains. In comparison studies of these two transport codes in identical geometries, they agree in the 10% to 30% range in total neutron and photon rates. In the past 8 months, Ian Dawson (University of Sheffield), has begun contributing FLUKA results on ATLAS background rates, concentrating on the "100 MeV" trigger problem in the muon system. He is also turning to the issue of activation. But the complexity of FLUKA geometry specification prevents him from rapidly cycling the numerous geometry options needed for engineering tradeoff studies.

ATLAS is at a critical juncture where the engineering development of the radiation shielding is proceeding rapidly, and there are many design options to consider. At the same time, ATLAS collaborators involved with various subsystems are in need of more detailed information on radiation levels or particle rates and spectra. Their concerns cover the whole range from damage to detector, electronic, or data link components to trigger rates induced by intermediate energy charged particles.

Whether done with Geant3/GCALOR or Milan FLUKA, these calculations are extremely CPU intensive. A 1000 event GCALOR run for one option study takes four to six days with a dedicated 400-500 MHz CPU. Exploring the parameter space of shielding design options takes many such studies. In addition, there is a growing demand for flux maps divided into energy decades, for spectra at various locations, and for spectrum output files. These types of runs take factors two to three longer than standard runs. Finally, the worst problem confronting the trigger community right now – the "100 MeV" charged particles in the muon trigger system – will require 10's to 100's of thousands of events to understand, in side by side comparison runs of GCALOR and FLUKA. Many minimum bias events must be run to develop sufficient statistics on this 100 MeV sample to understand its primary sources.

Until recently the CPU power available for these calculations was minimal, allowing calculation of only a few options at points "chosen" in the complex parameter space. But the appearance of the Tier 1 Center at Brookhaven is already making a difference. Since mid-September 1999, four 500 MHz dual processor machines at Brookhaven (linux001-004) have been dedicated to the production of ATLAS background calculations. These machines enabled the the calculation of eleven geometry options for the Shielding Engineering meeting at CERN on November 8th, 1999, and are yielding more than 25 options in time for the February 4th, 2000 meeting. What is more, there are 10 more linux machines coming online very soon, and these will allow the calculation of particle spectra, 1 MeV Equivalent Neutron damage rates, single event upset rates, and single event burnout rates in detector and electronics components of ATLAS.

More generally, within U.S. ATLAS the background calculation work will soon be benefiting from additional manpower. Over the years, Sue Willis and Vladimir Sirotenko at Northern Illinois University have made a number of solid contributions to the understanding of ATLAS backgrounds using Geant3/GCALOR on DEC Alpha machines at N.I.U. Just recently Willis has acquired new linux PC hardware, and is jumping into the engineering options calculations of the current era.
This backgrounds calculation project is mission critical to the success of ATLAS, and has evolved in the past year and a half into an activity in which the U.S. ATLAS community is making a major ongoing contribution in the area of detector engineering design development.

Personnel involved: M. Shupe, University of Arizona; V. Sirotenko, S. Willis, Northern Illinois University.

4 Collaborative Tools (WBS 2.2.3)

4.1 Syncomat

The University of Michigan plans to build on their successful web-based lecture tools (Syncomat-3000) in support of the overall ATLAS and US-ATLAS software training activities. Syncomat has been released and has been undergoing extensive testing in collaboration with CERN. Many LHC software training talks have been made available to ATLAS and CMS. Unique features include high quality audio and clear PowerPoint slides presented together with a synchronized video image of the speaker.

We will also enhance existing video conferencing tools through the development and implementation of Quality-of-Service techniques in collaboration with Internet2/CERN and MERIT network researchers. We will continue to aid in the expansion of CERN/US networking bandwidth through efforts with UCAID and CERN networking.

Personnel involved: Homer Neal, Bob Ball, Shawn McKee

4.2 DOE2000 Collaboratory Tools

The DOE2000 Collaboratory Program has developed a number of tools and underlying technologies to support network-based collaboration. U.S. ATLAS will work with the developers to incorporate those tools into the ATLAS experiment. The tools of interest include enhancements to the video conferencing system and Web-based electronic notebooks. DOE2000 has also developed mechanisms to ensure priority service on the Internet (Differentiated Services) and has implemented a distributed security model for authentication and authorization.

The DOE2000 work on video conferencing has improved the tools developed for the MBone and has implemented a remote camera controller so that users who are watching meetings from their home sites can change cameras or pan/tilt/zoom camera to improve the image they see. We will work with the VRVS Group to incorporate these improvements into the system that ATLAS already uses for most of its meetings. The electronic notebooks are another new tool that will become very useful as the distributed collaboration begins to work on testbeams and detector commissioning. The notebooks provide a shared document where collaborators text, images, video and other media.

The security and Quality 0f Service mechanisms will significantly enhance the controls and performance of collaborative applications. We will be working with the Grid developers who are incorporating these technologies into a collaboration environment that will also support shared data views and analysis control. We will investigate whether it is possible and appropriate to integrate the ATLAS Analysis Framework into the Grid collaboration framework.

5 Software Support for U.S. ATLAS (WBS 2.2.4)

Closely allied with the U.S. ATLAS Tier 1 Computing Facility at BNL will be a software support effort to provide current, tested installations of ATLAS offline software on ATLAS platforms, for use at the Tier 1 facility and at other U.S. facilities mirroring the Tier 1 installations. While the support for third party and community software employed by ATLAS – such as Objectivity and the LHC++ suite – will rest with the Computing Facilities organization, support for U.S. installations of the

ATLAS offline software itself will be a responsibility of the Software organization. Charging Software with this responsibility ensures that the support function rests organizationally most closely to both the expertise base of the ATLAS offline software and the community best able to provide oversight and set the program and priorities of the support operation.

The support function will include maintenance of U.S. installations of core offline software and subdetector specific software for all subsystems. A help-desk function primarily for U.S.-specific software installation and usage issues (with referral to the ATLAS help services for more general issues) will be included. Allocation of support effort and facility resources in terms of what versions are maintained in the U.S. on what platforms at what level of support will be determined by prioritizations set by the U.S. ATLAS offline software user community.

The support operation and its oversight must be closely coupled to the Tier 1 facility. The principal support person will be the Software Librarian. Oversight of the support function will be performed by a Software Support Coordinator reporting to the Software Manager. Both Librarian and Coordinator will have close interaction with Facility personnel and both should reside at BNL. The Software Support Coordinator should be an experienced and active software developer who can gather the support requirements and priorities of the U.S. community and accordingly prioritize and direct the work of support personnel.

6 Software Training in U.S. ATLAS (WBS 2.2.5)

Until the early part of 1999, the primary ATLAS programming language was Fortran, and the analysis structure was procedurally based. This was the kind of code used for ATLAS reconstruction and physics studies leading to the Physics Technical Design Report.

Beginning in the spring of 1999, there has been a concerted effort to move entirely into an object-oriented methodology with C++ as the primary language. This has been a major change for both U.S. and non-U.S. physicists, and has required an organized software education program for physicists both at CERN and in the U.S. This involves more than learning a new software language – much more important is the learning of Object-Oriented methods and design, which represents a new approach to software development and analysis structure.

In the United States we have organized a number of courses to date, with others planned. The course topics and locations were determined by a broad survey of the interests of U.S. ATLAS collaborators. The students of these courses have comprised 44 U.S. ATLAS physicists from 8 universities and 3 national laboratories. Two courses in object-oriented analysis and design were held in the summer of 1999, one at Brookhaven and one at the University of Chicago. Both were taught by professional OO/C++ software educators from Object Mentor Associates, a Chicago-based firm which has previously taught OO courses for BaBar and STAR physicists. A third hands-on course on Geant4 (the new OO/C++ version of Geant) was held at Fermilab and taught by Andrea Dell'Acqua, a CERN/ATLAS physicist who has been a leader in Geant4 development at CERN. The students included 18 U.S. ATLAS physicists and an additional 6 physicists from CDF, D0, and CMS, in order to help other collaborations gain experience with Geant4. Each of the courses has attracted a number of physicists from different institutions and working on different parts of the ATLAS analysis effort. A fourth course in Advanced Object-Oriented Analysis is being planned for March 2000.

Each of these courses has been an intensive week-long project, running from about 9:00 am to 5:00 pm each day. The time has been split roughly 50-50 between lectures and hands-on exercises, and including extensive discussions. Each of them has been well received by the students. In addition to these courses, a number of U.S. ATLAS physicists have learned C++ syntax on their own or through other courses, and have taken tutorials or courses in related software techniques or tools. The courses have provided both a common educational basis and a stimulus for other studies.

We believe the U.S. ATLAS software education program has been quite successful. Because it has involved a number of U.S. physicists taking the same courses, it has provided a common "jump-start" to OO technology and to continuing software education. Students have discussed OO ideas in class, and have then applied them to software design and program structure. The result has been a widespread OO-mobilization among U.S. ATLAS physicists. An added advantage is that software physicists from a number of different institutions on U.S. ATLAS have gotten to know each other better, and share a common educational background.

The extent of U.S. ATLAS participation in these courses is indicated on the enclosed chart listing the students, their institutions and subdetector involvement, and the courses they have taken (in red). The result of the program has been a significant improvement in the ability of U.S. physicists to develop object-oriented code and to participate in the development of ATLAS software.

Name	Institution	Group	Courses needed (X means wants to take)					Email
			OOAD1	C++	AdOOD	Geant4	Java	
Hinchliffe, Ian	LBL	*Physics						I_Hinchliffe@lbl.gov
Protopopescu, S.	BNL	Core	Done	Expert	March?			serban@d01.phy.bnl.gov
Tull, Craig	LBL	Core	Expert	Expert	X	X	X	cetull@lbl.gov
Malon, David	Argonne	Core, Tilecal	Expert	Expert	March?	11/8/99	X	malon@anl.gov
Gilchriese, M.	LBL	ID						gilg@lbl.gov
Calafiura, Paolo	LBL	ID/ST			X	11/8/99		PCalafiura@lbl.gov
Siegrist, Jim	LBL	ID/ST	X	X	X	X	X	jlsiegrist@lbl.gov
Vacavant, L	LBL	ID/ST	Taken	Taken	X	Done	X	L_Vacavant@lbl.gov
Assamagan, Ketevi	Hampton Univ	ID/TRT	x	Done		Done		Ketevi.Adikle.Assamagan@cern.ch
Baker, Keith	Hampton Univ	ID/TRT	8/9/99 B			11/8/99		baker@cebaf.gov
Luehring, Fred	Indiana	ID/TRT	9/21/99 Ch	Done		11/8/99		fred@oolitic.physics.indiana.edu
Manara, Andrea	Indiana	ID/TRT		Done		Taken		Andrea.Manara@cern.ch
Vassilakopoulos, Vass	Duke	ID/TRT		Done		11/8/99		vassilis@phy.duke.edu
Lanni, F.	BNL	LAr	8/9/99 B		X	9/20/99 C		flanni@bnl.gov
Ma, H.	BNL	LAr	8/9/99 B		X	?		hma@bnl.gov
Rajagopalan, Srini	BNL	LAr	8/9/99 B		March?	9/20/99 C		srinir@sun2.bnl.gov
Stumer, I.	BNL	LAr	8/9/99 B		X	9/20/99 C		stumer@bnl.gov
Vanyashi, Sasha	BNL	LAr	8/9/99 B		March?			vanyashi@rcf.rhic.bnl.gov
Dodd, Jeremy	Columbia	LAr	8/9/99 B					dodd@nevis1.nevis.columbia.edu
Leltchouk, Mikhail	Columbia	LAr	8/9/99 B	?	X	7/19/99 C		leltchou@nevis1.nevis.columbia.edu
Parsons, John	Columbia	LAr	8/9/99 B	X	X	X		parsons@nevis1.nevis.columbia.edu
Seligman, Bill	Columbia	LAr	8/9/99 B	X	March?	11/8/99		seligman@nevis1.nevis.columbia.edu
Seman, M.	Columbia	LAr	X	X	X	X		seman@nevis1.columbia.edu
Loch, Peter	Univ of Arizona	LAr	8/9/99 B			X		loch@physics.arizona.edu
Nevski,Pavel	BNL	Muon	Done	Done				nevski@bnl.gov
Wenaus, Torre	BNL	Muon	Done	Done	March?	11/8/99		wenaus@bnl.gov
Shank, Jim	Boston University	Muon	8/9/99 B		March?	Taken		shank@bu.edu
Diehl, Ed	Univ of Michigan	Muon	9/21/99 Ch		X	X		diehl@umich.edu
Han, Chunhui	Univ of Michigan	Muon	9/21/99 Ch	Done				chunhuih@umich.edu
Hou, Suen	Univ of Michigan	Muon	9/21/99 Ch			X		suen.hou@cern.ch
Levin, Dan	Univ of Michigan	Muon	9/21/99 Ch	X	X	Taken	X	dslevin@umich.edu
McKee, Shawn	Univ of Michigan	Muon	9/21/99 Ch	X	March?	Taken	X	smckee@umich.edu
Xu, Qichun	Univ of Michigan	Muon	9/21/99 Ch					xup@umich.edu
Zhou, Bing	Univ of Michigan	Muon	x	?	X	X		bzhou@umich.edu
Adler, S.	BNL	Muon?	8/9/99 B					adler@ssadler.phy.bnl.gov
Stratos, E.	BNL	Muon?	8/9/99 B	X	March?	X		stratos@bnl.gov
Gunter, David	Argonne	Tilecal	9/21/99 Ch	Done				gunter@mcs.anl.gov
LeCompte, Tom	Argonne	Tilecal	X	?	March?	11/8/99		lecompte@anl.gov
May, Ed	Argonne	Tilecal	9/21/99 Ch		X	11/8/99	X	may@anl.gov
Price, Larry	Argonne	Tilecal	x	X		X		lprice@anl.gov
Wagner, Bob	Argonne	Tilecal	9/21/99 Ch	Done	X	11/8/99		rgwcdf@anl.gov
Anderson, Kelby	Univ of Chicago	Tilecal	9/21/99 Ch	X	X	X		kelby@hep.uchicago.edu
Carcassi, Gabriele	Univ of Chicago	Tilecal	Done	Done	March?	11/8/99		carcassi@hep.uchicago.edu
Gupta, Ambreesh	Univ of Chicago	Tilecal	9/21/99 Ch	Done	March?	11/8/99		agupta@hep.uchicago.edu
Merritt, Frank	Univ of Chicago	Tilecal	9/21/99 Ch	8/9/99 F	March?	11/8/99		merritt@hep.uchicago.edu
Oreglia, Mark	Univ of Chicago	Tilecal	9/21/99 Ch			11/8/99		oreglia@uchicago.edu
Pilcher, Jim	Univ of Chicago	Tilecal	9/21/99 Ch	X		11/8/99		pilcher@hep.uchicago.edu
Blair, Bob	Argonne	Trigger				11/8/99		reb@anlhep.hep.anl.gov
Schlereth, Jim	Argonne	Trigger	X	Expert		11/8/99		jls@hep.anl.gov
Abolins, Maris	MSU	Trigger	X	X	X	X		abolins@pa.msu.edu
Brock, Ray	MSU	Trigger						brock@msupa.pa.msu.edu
Hauser, Reiner	MSU	Trigger	Expert	Expert				
Pineiro, Beatriz	MSU	Trigger		Expert				
Pope, Bernard	MSU	Trigger	X	X	X	X		pope@msupa.pa.msu.edu
Zobernig, Haimo	Wisconsin	Trigger						haimo.georg.zobernig@cern.ch
Name	Institution	Group	Courses needed (X means wants to take)					Email
			OOAD1	C++	AdOOD	G4	Java	