

*ATLAS Muon Discussion Note***The ATLAS Muon Spectrometer****Database Task***Draft 1.0**12 November 1999**S. Goldfarb, University of Michigan***1.0 Introduction**

This document outlines the current status, future plans and long-term goals of the ATLAS Muon Spectrometer Database Task. It is intended to serve as a working document for software developers wishing to contribute to the task, as well as an initial identification of the services to be offered by the task to the muon spectrometer community in the coming years.

One might argue that it is too early to specify the final form of the muon software. Indeed, the Architecture Task Force (ATF) has only recently presented the collaboration with a document stating its own vision for the overall framework. This plan will clearly have a major bearing on the design of the muon software. It is vital, however, that the software development remain closely bound to the needs of the detector communities. For the muon system, many of these needs can be identified today, such as the development and testing of the simulation and reconstruction programs, and the provision of software for test beam studies. It is imperative that we proceed now with a strategy that will allow us to address these needs, while pressing ahead to achieve the goal of a complete, robust and maintainable software base in the long term.

With this in mind, we define several short-term tasks for the muon database software development, describe plans for completing those tasks and provide estimates of the necessary resources. In addition, we present a broad outline of the basic functionality to be developed during the coming years and describe an overall strategy to reach this goal. This does not pretend to be an exhaustive list of the necessary services nor a detailed description of the implementation, but rather a first iteration in an evolutionary process. In the interest of providing a concise, working document, emphasis is placed on the task management, in terms of time and resources, rather than on the technical aspects of the database.

**2.0 The Scope of the Task****2.1 The Data Stores**

The main purpose of the muon database task is to provide efficient access to muon spectrometer data in the manner required by the various software applications, including event simulation

reconstruction and analysis. The data to be accessed can be classified into a few general categories, independent of its actual physical organization. The two classifications we will address here are the *event store* and the *detector description store*.

The event store contains information describing physics events, such as raw data, generator tracks and reconstructed objects. The detector description store contains the information necessary to describe the state of the detector to the required level of detail at a given time. This includes the geometrical description, alignment and calibration data, and relevant detector construction information.

## 2.2 Data Access

Access to the data is provided through a transient interface to the application software. It is preferable that this interface be stable and simple to use by a variety of applications throughout the life of the experiment. That is, changes to the underlying database technology should not require major modifications to the application software in order to maintain the same functionality.

The interface comprises at least one layer of abstraction between the data and the user application. This abstraction involves the extraction, interpretation and translation of the data into C++ objects which are independent of the application. This is referred to as the *generic model*. The basic idea is to provide objects with sufficient generality that they can easily be mapped to the requirements of the application software. In many cases, software which is independent of the detector-type can be used at this stage, introducing a second level of generality and possibly reducing the size of the task, in terms of manpower and resources.

Following the data extraction, an application-specific and/or detector-specific interface is layered on top of the generic model. Construction of such an interface requires knowledge of the input requirements of the application program (e.g. GEANT4 physical volume parameters) as well as an understanding of the objects made available by the generic model (e.g. geometrical dimensions of an MDT multilayer).

Event data will be coupled to the detector data through the use of logical identifiers. These identifiers are a sequence of integer indices specifying, in a hierarchical manner, the active detector component to which the signal is attached. As an example, an identifier for an MDT digit would contain integers specifying: side, ring, station, multilayer, layer, tube. All sensitive detector elements must be uniquely and unambiguously identified.

## 2.3 Data Storage

The database interface must also provide a mechanism for the storing of user-defined objects in the database. These will include reconstructed objects, analysis objects, corrected data objects or even various versions of the detector description for Monte Carlo studies. We will not address the details of such a mechanism here, as much of its implementation will be developed independently of the detector subsystem and will be provided as common software.

## 3.0 Implementation for the Muon System

### 3.1 The Detector Description

#### 3.1.1 AMDB

The Muon Spectrometer software currently benefits from the existence of a persistent detector geometry database, in the form of formatted ascii file called **AMDB**. This file contains a minimal set of parameters necessary to build the geometry of the four muon subsystems: MDT, RPC, TGC and, more recently, CSC, as well as several of the (non-active) service volumes, such as the toroid. Translation of these parameters to a geometrical description requires the use of parsing and interpreting software. The software was written in a manner which easily allows for modifications and additions to the geometry, although the interpretation of these changes must be added to the application code.

Perhaps the most important advantage of the AMDB package is the fact that it places the geometrical parameters into one human-readable (and editable) file. Ideally, this would make it possible for all software applications depending on the geometry to input the data from one single source, preventing the existence of multiple, possibly differing, representations of the detector in the software. Unfortunately, this is not the case, for several reasons.

First of all, while the format of the ascii file is fixed and fairly well-documented, interpretation of the parameters is still left to the interface software. The existence of more than one interface to AMDB can lead to differing geometries if the parameters are mistakenly interpreted by one or more different methods. Furthermore, the practice of maintaining only a minimal set of parameters tends to increase this possibility, because a significant number of calculations are required to build the description and because there is little possibility for cross-checking.

While these difficulties can be overcome by the required usage of a single package of interpreting routines, the resulting software must contain an important degree of detector-specific knowledge. The division of intelligence between the software and the data requires a significant maintenance effort and is vulnerable to error during modification to the description. Furthermore, the lack of structure to the data representation makes it more difficult to provide mechanisms to navigate the hierarchy of the geometry.

#### 3.1.2 AGDD

The development of an ascii detector geometry description in the eXtensible Mark-up Language (**XML**) has recently been proposed for all of the ATLAS subsystems by a small working group lead by Stan Bentvelsen. Like HTML, XML is a markup language which is used to format an ascii file through the use of tags. Unlike HTML, XML is actually a metalanguage which can be used to define other markup languages, through the use of a Document Type Definition (**DTD**) file. The DTD describes the structure of the XML files by listing the available elements, defining their attributes and specifying which other elements they may contain (substructure).

XML is a standard, defined by the World-Wide Web Consortium (W3C) in the Spring of 1998, and is commonly expected to replace, or at least co-exist with, HTML in the near future. As such, one can expect it to have a relatively long lifetime and to receive the support of numerous commercial vendors. Although XML 1.0 has only been around for a short time, there already exist a variety of commercial tools for editing, parsing and viewing XML files.

The DTD defined by Bentvelsen and Marc Virchaux can be found in the software repository at [offline/DetectorDescription/AGDD/data/AGDD\_1.04.dtd]. It defines elements associated to general geometric parameters (e.g. solids, volumes, compositions) which are independent of the type of detector being described. Christian Arnault has provided a mechanism for parsing XML files based on the DTD to create generic (non-application specific) transient C++ objects. This is complementary to the detector-specific model currently defined in the offline software [offline/MuonSpectrometer/MuonDetDescriptor]. The combination of the DTD and the new generic model are currently referred to as the ATLAS Generic Detector Description (AGDD), but are a part of the DetDescriptor package.

Perhaps the greatest strength of XML is that it provides a mechanism for structuring the data. Hierarchical descriptions for the MDT and RPC geometries already exist in the Detector Description software and can be easily mapped to the XML file structure, greatly simplifying the interface and reducing the risk of error in the interpretation of parameters. Furthermore, this hierarchical structure maps naturally to a GEANT-style detector description. Bentvelsen has already developed a general application, called **G4builder**, to build GEANT4 objects from the C++ objects constructed in Arnault's model from the XML. That is, the full path from persistent ascii description through the generic model to GEANT4 can now be achieved by building an XML description and then running the G4builder application.

In addition, Virchaux has updated his visualization program (*persint*), which is capable of displaying geometries read from AMDB, to now display geometries described in the AGDD XML files. So, several important tools for development are in place.

### 3.1.3 Potential Pitfalls

While the development of geometrical models in the AGDD framework appears feasible for the direct construction of GEANT4 geometries, it is doubtful that a completely general interface, such as G4builder, can provide all the information needed by the simulation. It is more likely that detector-specific interfaces must be developed to take into consideration issues such as the parametrization of volumes and the details of the sensitive detector information.

From the point of view of reconstruction, logical identifiers will be used to associate event data with the active components of the detector. A fairly primitive device, based on the relative  $(r, \phi, z)$  location of a detector component within its mother volume has been built into AGDD for the automatic generation of identifiers. Use of this mechanism to construct unique identifiers based on the hierarchical structure is not clear. For example, the current scheme for the RPC system labels doublets, layer-pairs, layers and strips. Moreover, the use of an automatic identifier generation mechanism could pose other difficulties, such as the accidental renumbering of a component due to the addition or removal of a different component. Serious consideration must be given as to whether a robust mechanism could or should be built into AGDD or provided by another means altogether.

Finally, it is important to note that the muon community currently benefits from the existence of working well-tested software based on AMDB. This software, including the *muonbox* reconstruction program, relies on the existing AMDB interface, to which it is strongly coupled. The importance of maintaining a single Detector Description database implies that AMDB would have to be phased out if AGDD is phased in. Any proposal to replace the AMDB interface, how-

ever, must take into consideration the repercussions and plan an evolutionary development strategy, so as not to create a major break in the working code.

### 3.1.4 Evolutionary Development Strategy

The XML files are considered, at least for now, by the ATLAS Database Group as a possible interim (2-3 year) solution for the persistent storage of the Detector Description. This, of course, depends on feedback from the detector communities as to the feasibility and usefulness of the conversion. For the muon system, this would involve an evaluation of the complete conversion of AMDB to the XML format, in terms of manpower requirements and gained functionality. Such an evaluation will require a coordinated effort between the subsystem experts possessing detailed knowledge of the geometrical detail of the detector components, software experts with a clear understanding of the current representation of those details in AMDB and potentially new developers interested in building the new XML description.

In order to preserve the current working AMDB interface, it is proposed that the first evolutionary step be to essentially convert AMDB version M2.8 in its current form to XML. No further detail would be added to the description, with the inner structures of the components represented by a simple sequence of parameters, as is currently the case with AMDB. The result of such a study may help to locate difficulties in the conversion process itself and will help to maintain the working software chain, as only minimal changes to the existing interfaces will be required. It will not, however, serve as a sufficient test for evaluating the potential gains in functionality, as the data will still require the existence of the current interpretation software to be interfaced to the application programs, such as muonbox and the GEANT4 simulation. This software must eventually be decoupled from the application programs to the generic description in order to ensure the existence of a single interpretation.

A second evolutionary step may be taken shortly after or even simultaneous to the first. In that step, the geometry described in AMDB version M2.8 would be converted to the XML format, as above, but based on a complete hierarchical description of the geometry, down to the lowest level of detail allowed by the AMDB data. This step will require a clear interpretation of the AMDB data and will require the participation of the subsystem (MDT, RPC, TGC, CSC) experts. The more complete description, however, will allow for an evaluation of the potential to uncouple the interpretation software from the application packages, to evaluate the utility of the commercial XML parsing tools and to examine the effort required to build the GEANT4 interface. Any final decision to completely convert the persistent description to XML can only be made by the muon community following a complete evaluation of the costs in manpower and resources against the potential benefits.

## 3.2 The Event Store

### 3.2.1 Testing Mechanism

Access to the simulated event data is currently provided through the input of ZEBRA files stored on disk or staged in from magnetic tape. For the purpose of software testing, a simple event loop, providing access to GEANT3 digits, called `test_event`, is included in the offline software (`offline/Event/EventManager/test`). This program has been used to test the MDT and RPC Detector Description software by checking the encoding and decoding mechanisms of the logical identifiers and by ensuring that the positioning methods of the geometrical models agree with the GEANT3 transformations associated to the simulated digits. `Test_event`

is essentially the precursor to PASO and will be phased out soon, with the testing software being ported to the latter framework.

### 3.2.2 PASO

A provisional C++ framework, called **PASO**, has been recently added to the offline software (`offline/Applications/Paso`) for user development of data reconstruction and analysis software. Although the user is provided with C++ data objects for manipulation with her/his own software, PASO inputs data using a wrapped SLUG interface to the ZEBRA tapes. It was proposed by the Database Group at the conclusion of the October 1999 workshop to look into providing data through an interface to an Objectivity/DB data store.

Although PASO is considered provisional, pending the recommendations of the ATLAS Architecture Task Force (ATF), access to the event store can be expected to follow a similar scheme, based on the existing Event Model. As with the Detector Description, a layer of software will insulate application programs from the persistent data. This software will provide the necessary interface to simulation hits and digits, raw data and reconstructed objects. The exact extent of the insulation will depend on compromises between data access optimization and the need to be independent of the persistent database technology.

## 3.3 Production Databases

Data pertaining to the construction and testing of detector components is currently being gathered in databases private to the various muon subsystems. It can be expected that access to at least some of this data will be desired during data reconstruction and analysis and perhaps for the simulation of realistic Monte Carlo data samples. An example would be the identification of noisy trigger channels or broken wires in the MDT. Some of the data may also be of value in the future for unforeseeable circumstances.

Detector production information potentially needed by the offline software, should be integrated into the Detector Description data store. Toward this effort, it will be necessary for the muon community to identify parameters which will require immediate access, those which may require eventual access and perhaps a few intermediate scenarios. This does not imply an immediate decision on the implementation of the persistent database, but rather recommends the definition of foreseeable requirements.

For production data to be accessed, it will be essential to create a mapping between the logical detector element identifiers described above and the physical construction identifiers (i.e. bar codes) and/or to other values used as keys in the production databases. The exact implementation of this mapping is not yet defined, but will be kept independent of both sets of data.

## 3.4 Calibration and Alignment Data

Both calibration and alignment data are also expected to be an integral part of the Detector Description data store. Access to the data may be direct or through corrections to existing "perfect" Detector Description. This will depend on the needs of the application software. Requirements for the integration of both calibration and alignment data into the Detector Store warrant further investigation in the coming year.

## 4.0 Organisational Issues

### 4.1 Task Coordination

The Muon Database Task Leader takes on the responsibility of charting a course for the database software development in coordination with the Muon Reconstruction and Simulation Task Leaders, under the guidance of the Muon Software Coordinator (Table 4-1). The task leader is also responsible for coordinating the implementation of this course of action. This involves the communication and agreement of the overall strategy within the muon community, the coordination of development with the ATLAS Database Group and the other subsystem task leaders, the identification of necessary actions and milestones, the quantification of resources needed, the provision of training and/or example software, and the development of general database software for the muon system.

**Table 4-1** The tree-structure organization of the Muon Software Group.

Software (general)		
	Gilbert Poulard	
<b>Muon Simulation</b>	<b>Muon Database</b>	<b>Muon Reconstruction</b>
Adele Rimoldi	Steven Goldfarb	Jean-Francois Laporte

### 4.2 Communication

Several forums exist to facilitate in the communication of Muon Database issues:

- Muon Spectrometer Database Home Page:  
`http://home.cern.ch/muondoc/software/Database/`
- Muon Software Listbox:  
`atlas-muon-sw@atlas-lb.cern.ch`
- ATLAS Database Home Page  
`http://atlasinfo.cern.ch/Atlas/GROUPS/DATABASE/database.html`
- ATLAS Database Listbox  
`atlas-database@atlas-lb.cern.ch`
- ATLAS Detector Description Listbox  
`atlas-detector-description@atlas-lb.cern.ch`

In addition, I can be reached directly via e-mail at <Steven.Goldfarb@cern.ch>.

### 4.3 Resources

Manpower and resources for the Muon Database task come from within the muon community. This includes (at least part-time) dedicated software developers, as well as persons involved in the construction and testing of the detector who have a strong understanding of the qualities of the various subsystem elements. In most cases, this will involve a coordinated effort between these groups. The identification of people wishing to take responsibility for various database

software tasks should be recommended to the task leader to allow for proper coordination and to avoid the loss of valuable resources due to the unnecessary repetition of work.

## 4.4 Timetable

Key muon database development issues have been identified and divided into two categories of relative completion time: Short Term Tasks (half year) and Long Term Goals (up to 5 years). The times are based on the existence of work in progress, expected duration of the projects and, most importantly, on the urgency of the completion of the project for the muon software efforts.

### 4.4.1 Short-Term Tasks (Up to Six Months)

The following projects are, most likely, already underway and require several months time for completion and testing:

1. Completion of AMDB “short-cut” interface to the GEANT4 simulation to allow for simulation software development with the same geometry as reconstruction.
2. Upgrade of AMDB to include recently added details to the RPC and CSC geometry.
3. Development of example geometries in AGDD for the four subsystems: MDT, RPC, TGC, CSC (a preliminary barrel station has already been defined).
4. Evaluation of AGDD as a replacement for AMDB and its various interfaces, including identification and training of responsible persons for the four subsystems.
5. Provision of feedback to AGDD development team for definition of a DTD which better satisfies the needs of the muon community.
6. Creation of interface from AGDD to the Muon GEANT4 Simulation software to test the full chain from XML.

### 4.4.2 Long-Term Goals (Up to Five Years)

The following are major software efforts which are foreseen to be needed for the completion and initial testing of the OO muon software:

1. Development of the Muon Detector Description Database, including:
  - single modifiable persistent data store;
  - generic model built from the persistent store and providing the application interface to the data;
  - a fixed, well-defined logical identifier scheme;
  - access mechanism for alignment, calibration and production data through the use of a mapping scheme linking detector element data to the logical identifiers.
2. Development of the Muon Event Database, including:
  - access to the raw data through a standard interface;
  - storage and retrieval mechanisms for reconstructed muon data and analysis objects.



3. Equivalent Detector Description and Event data access and storage schemes for test beam analysis.

## 5.0 Conclusion and Outlook

This document presents only a brief outline of the status, goals and strategies of the ATLAS Muon Database Task. Many details, in particular technical specifications of the database technology, have been purposely omitted for the sake of brevity and to maintain the focus on the managerial issues. In fact, decisions on the technical issues will often be taken at a higher level, based on input from all detector groups and on the needs of the entire experiment. Nevertheless, the task we face is an important one that requires careful planning if we are to achieve the goal of a complete, working, maintainable software base in only a few years' time.

As mentioned in the introduction, this document is meant to serve as the first step in an iterative process. The task leader welcomes any and all feedback pertaining to its content and to the task at hand. Please feel free to contact me at any time to discuss these issues.