ATLAS High-Level Triggers, DAQ and DCS

Technical Design Report

Issue:	Annotated Outline
Revision:	2
Reference:	ATLAS TDR-xx
Created:	12 November 2002
Last modified:	24 January 2003
Prepared By:	ATLAS HLT / DAQ / DCS Group

ATLAS Collaboration

CERN
European Laboratory for Particle Physics (CERN), Geneva

Acknowledgements

The authors would like to thank Mario Ruggier for preparing the template upon which this document is based and the DocSys group for their help in using it.

Table Of Contents

Part 3

Part 4

Part 1

Global View

1 Overview

1.1 Main system requirements

1.1.1 From physics

text

1.1.2 From performance (Read-out, selection)

1.1.3 Functional and operational

1.2 System functions

1.2.1 Detector R/O

1.2.2 Event selection/rate reduction

1.2.3 Movement of data

1.2.4 Storage of data (events, conditions, etc.)

1.2.5 Experiment Operation

1.2.6 Detector controls

1.3 Types of data TDAQ deals with

1.3.1 Detector control values

1.3.2 Event data

1.3.3 Configuration data

1.3.4 Conditions data

1.3.5 Statistics and monitoring data

1.4 Glossary

1.5 References

1-1

1-2

2 Parameters

This chapter is dedicated to the relevant parameters for the HLT/DAQ/DCS system. These include the detector readout parameters and the trigger selection for the correct dimensioning of the dataflow system and for understanding the data volumes that will need to be stored. These will be the subject of the first three sections.

Other important parameters for the correct definition of the system are the one coming from the monitoring requirements. These are discussed in the fourth section.

The last section is dedicated to the DCS parameters: the subdivision of the system in detector parts and the amount of configuration data traffic in case of cold configuration and re-configuration of possible faulty elements.

2.1 Detector R/O parameters

This section could be moved to Section 1.2.1, "Detector R/O".

The ATLAS detector is organized in sub-detectors:

- Inner Detector
    - Pixel
    - SCT
    - TRT
- Calorimetry
    - LAr Calorimetry
        - e.m. Barrel
        - e.m. EndCap
        - Hadronic EndCap
        - Forward Calorimeter
    - Tile Calorimetry
        - Barrel
        - Extended Barrels
- Muon Spectrometer
    - MDT
    - RPC
    - TGC
    - CSC

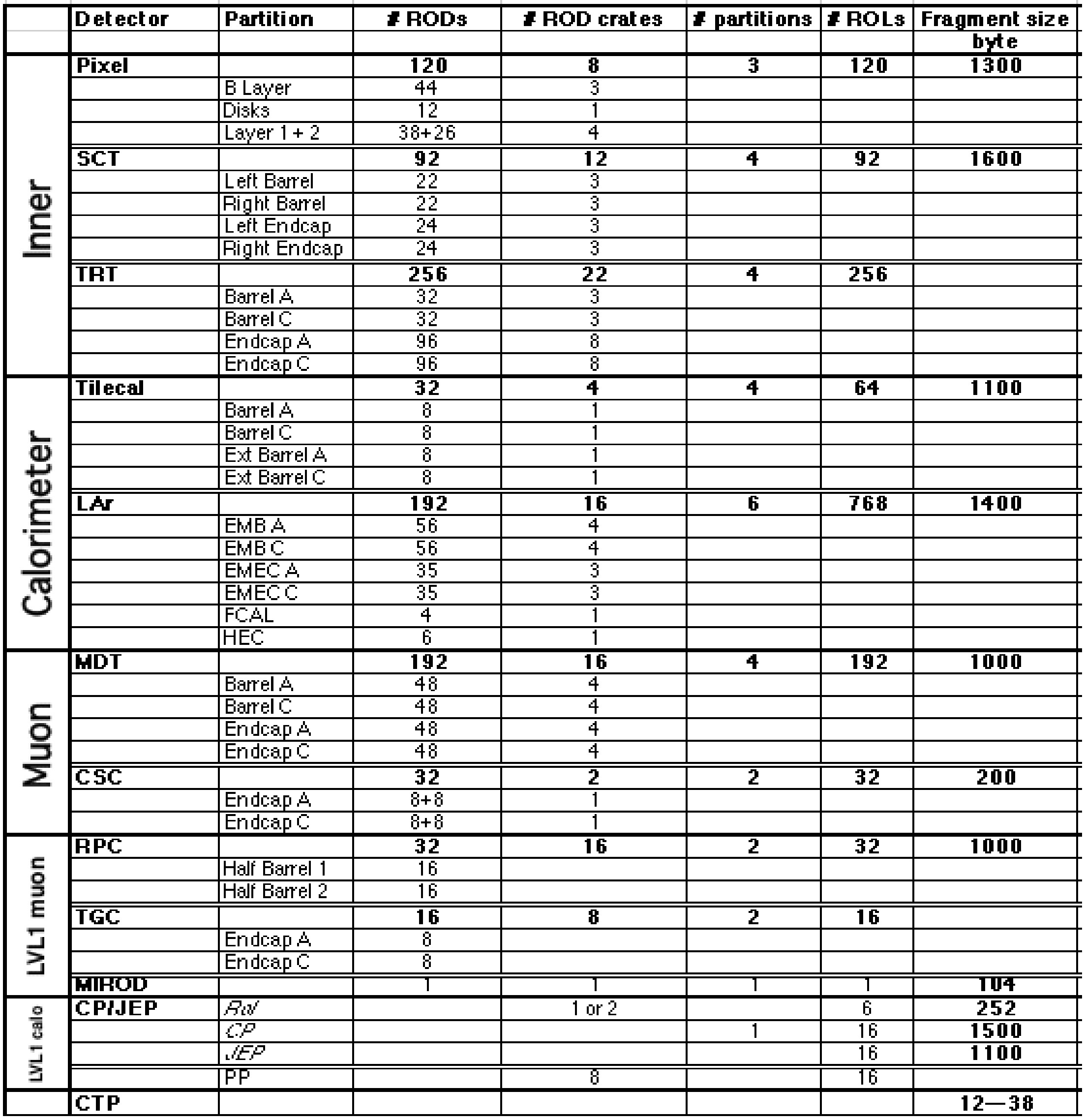
In terms of readout for signal to be transmitted to the Data Acquisition (DAQ) system, the LVL1 Trigger is another source of data and dedicated ReadOut Drivers (RODs) are used.

The organization in terms of readout is in fact slightly different and it is illustrated in the first sub-section, where a mapping of the ATLAS detector and trigger is specified in terms of data sources (the RODs) for the DAQ system in terms of the partitioning.

The concept of partition used throughout this chapter coincides with the TTC partition concept introduced by the LVL1 TDR.

2.1.1 RODs per detector per partition

text



	Detector	Partition	# RODs	# ROD crates	# partitions	# ROLs	Fragment size byte
Inner	**Pixel**		**120**	**8**	**3**	**120**	**1300**
		B Layer	44	3			
		Disks	12	1			
		Layer 1 + 2	38+26	4			
	SCT		**92**	**12**	**4**	**92**	**1600**
		Left Barrel	22	3			
		Right Barrel	22	3			
		Left Endcap	24	3			
		Right Endcap	24	3			
	TRT		**256**	**22**	**4**	**256**	
		Barrel A	32	3			
		Barrel C	32	3			
		Endcap A	96	8			
		Endcap C	96	8			
Calorimeter	**Tilecal**		**32**	**4**	**4**	**64**	**1100**
		Barrel A	8	1			
		Barrel C	8	1			
		Ext Barrel A	8	1			
		Ext Barrel C	8	1			
	LAr		**192**	**16**	**6**	**768**	**1400**
		EMB A	56	4			
		EMB C	56	4			
		EMEC A	35	3			
		EMEC C	35	3			
		FCAL	4	1			
		HEC	6	1			
Muon	**MDT**		**192**	**16**	**4**	**192**	**1000**
		Barrel A	48	4			
		Barrel C	48	4			
		Endcap A	48	4			
		Endcap C	48	4			
	CSC		**32**	**2**	**2**	**32**	**200**
		Endcap A	8+8	1			
		Endcap C	8+8	1			
LVL1 muon	**RPC**		**32**	**16**	**2**	**32**	**1000**
		Half Barrel 1	16				
		Half Barrel 2	16				
	TGC		**16**	**8**	**2**	**16**	
		Endcap A	8				
		Endcap C	8				
	MIROD		1	1	1	1	**104**
LVL1 calo	**CP/JEP**	*RoI*		1 or 2		6	**252**
		CP			1	16	**1500**
		JEP				16	**1100**
		PP		8		16	
	CTP						**12—38**

2.1.2 Fragment sizes per detector

Includes physics and calibration data.

Should have average values, spread and uncertainties; should be shown against luminosity; and against data compression schemes.

The fragment sizes reported in the previous table are indicative and they have to be seen as the maximum achievable figures.

Investigations are ongoing to obtain more realistic numbers for physics and calibration operations to resolve discrepancies with the values used in the Paper Model. The Detector people have to be contacted and an agreement on the numbers has to be found, based on the latest simulation they have for the sub-detector readout.

2.2 Trigger parameters

2.2.1 LVL1 rates

Three trigger menus: low luminosity without deferrals, low luminosity with deferrals and high luminosity (NB: low luminosity = 2 10^33).

Reference: Menus presented by Sefan Tapprogge in December T/DAQ week, http://doc.cern.ch/archive/electronic/other/agenda/a021484/a021484s3t3/transparencies/tdaq1202lvl1.pdf.

5 kHz of “Other items” not taken into account.

2.2.2 Parameters relevant for LVL2 processing

RoI sizes as defined in paper model document December 2001, ROB mapping (reference to back-up document, LVL2 processing strategy (B-physics !), reduction factors for sequential processing steps, LVL2 processing requirements.

2.2.3 Parameters relevant for Event Builder and Event Filter

Event building rate, Event Filter reduction factor, Event Filter processing requirements

2.3 Data rate summaries

LVL2 RoI request rates, data volume to LVL2 and EB per ROL, per ROS, total; for low and high luminosity.

2.4 Monitoring requirements

A more deep harmonization with the Monitoring chapters has to be achieved.

What has to be understood is if the requirements in terms of bandwidth have to be included in this section or whether they are described in the Monitoring chapter.

Again a contact with the sub-detector people is ongoing to understand from the Detector point of view what is the fraction of data we are speaking about and when and how frequently is the monitoring activated.

Thinking also about empty bunches in which most detectors would like to run dedicated calibration procedures to be monitored.

2.5 DCS parameters

numerology:

- how many channels you have in total. Where the channel concept has to be properly defined

- how many channels per detector. A table where this entry is specified per detector, together with the other values would be very useful, but we could put it together later.

- how many channels per partition, if that makes sense for DCS, if not why...

- what is the amount of data that you will need to configure

- if there is an error during running, would you re-configure only part of the system? Do you have a concept of cold and hot start?

- how many data you need to write? where? with which speed?

2.6 References

2-1

2-2

3 Operationial requirements for the TDAQ system

Why this chapter? While Chapter 2, "Parameters", somehow says what is required from TDAQ in terms of performance, this chapter should highlight what is expected by TDAQ when it is “used”.

The chapter contains the description of:

a. how an event is identified, at different levels in TDAQ.

b. What are the global TDAQ states. What are the detector and machine states. Including how TDAQ states relate to the detector and machine states. The global state transition

c. The defition of a run. How runs are identified. How an event is uniquely identified throughout the life of ATLAS. Types of runs. The question of the transition between runs. What is allowed during a run, what is done outside the run.

d. Partitioning: definition, operations

e. The general strategy to react to faults and errors (in TDAQ but also, and mainly, caused by external systems, such as the detector).

f. The role of data bases, what kind of data is permanently stored for what purpose (and where?).

3.1 Event identification

The need to identify (e.g. the ROB needs to store / retrieve events) an event at different levels in TDAQ (e.g. EL1ID in the ROB), how this is done (e.g. 32bit EL1ID in the fragment header), who sets the identification (e.g. the ROD). From the ROD down to mass storage.

Material from [3-1].

3.2 TDAQ states

We define 1) the DAQ states, 2) the detector (i.e DCS) states (relevant to TDAQ) and the machine states relevant to TDAQ. Followed by the global TDAQ state machine

3.3 The run

A lot of material may come from [3-1].

3.3.1 Generic definition

Material from [3-1].

3.3.2 Run Number

Definitin, purpose, where it is set into the event, who does it, how it is done. We should finally agree on the question of the run number set by the ROD.

3.3.3 Physics and calibration runs

Define what they are: what is their purpose, the actors (i.e. what a run type needs), where output goes.

Backup document on use cases?

3.3.4 Transition between runs

Material from [3-1].

We should settle the question of the "checkpoint".

3.4 Partitions and related operations

Defintion of what a partition is: what for, who are the actors participating to the partition.

Allowed partitons (here we should make reference to the constraints imposed by the TTC system and include the table of the detector partitions).

What can be done with partitions: join and split.

Material from [3-2].

How partitioning is realised on the system is reserved to Chapter 5, "Architecture" (and possibly Part 2 System Components).

3.5 Operations outside a run

Define what are the operations allowed when a run is stopped or when LHC is off. “Define” should include: the purpose of the operation, the actors, the expected result, the effect on TDAQ.

A backup document with use cases would be useful (if can be done).

This section was moved wrt the original layout so as to come AFTER the section on partitioning (since some of the operations might be done on partitions). It was also promoted one level up in the section hierarchy

3.6 Error/Fault reporting/handling strategy

Brief description of the global strategy here as the details are in Chapter 6. Emphasis should be given to 1) what TDAQ does when an internal error happens and 2) what TDAQ does when a fault happens outside TDAQ (but the fault affects the operation of the system).

3.7 Data Bases

What has to be stored permanently? at least give some broad categories and the source of the data. Then list what is the required functionality of the data base system(s). For example data related to configuration, conditions, monitoring, etc.

Material from this section should come from the efforts going on to collect requirements on data bases.

3.8 References

3-1 GIWG. Run and States

3-2 GIWG. Partitioning

4 Event selection strategy

4.1 The approach

HLT as a coherent entity, use of complementary features of LVL2 and EF (if not discussed before)

emphasis on inclusive signatures, have more refined tools (other selection algorithms, more exclusive / topological criteria etc) at hand

4.2 Selection objects

define (physics oriented) objects (e.g. e, mu, ...) to be used for the selection, describe in the following sub-sections the high-level (algorithm) steps to define candidate objects

4.2.1 Electron/photon

4.2.2 Muon

4.2.3 Tau/jets/$E_T$miss

4.2.4 b-tagged jets

4.2.5 B-Physics

4.3 Trigger menus

define the basic trigger menu(s), covering the major part of the physics program

need to address various scenarios

4.3.1 Physics triggers

unprescaled signatures go here

4.3.2 Pre-scaled physics triggers

4.3.3 Monitor and calibration triggers

4.4 Physics coverage

describe the coverage (essentially impact of thresholds) on various physics processes of interest

4.5 Determination of trigger efficiencies etc.

4.6 References

4-1

4-2

5 Architecture

The purpose of the chapter is to describe the top level architecture of TDAQ, in terms of

- its place with respect to the other parts of ATLAS, as well as systems and services external to ATLAS,
- how the system is organised: functionally, in terms of sub-systems and in terms of more abstract elements,
- a generic architecture with a a definition of the abstract components that are visible at the architectural level
- how sub-systems map onto the generic architecture (“views”).
- how the scalability and partitioning can be performed and

finally it proposes a baseline architecture expressed by the realisation of the abstratc components.

Is DCS part of the architecture? Yes, but it is considered, as regards this chapter, as a black box with interfaces to TDAQ and external systems. The internals of DCS do not belong to this chapter.

5.1 TDAQ context

5.2 Context Diagram

A context diagram indicating what TDAQ interacts with. Should include a definition of the context elements (Probably could re-use what is in the TP).

5.2.1 TDAQ Interfaces

Here we detail how the context elements and TDAQ interface; we also define what data is exchanged (who is generating it, who is using it).

An interface is defined in terms of the partners (in TDAQ and outside TDAQ), who is responsible for the interface (within TDAQ and outside TDAQ), what data flows in/out of the intreface and where the interface is documented. It is proposed to split things in 2 parts: I/F internal to ATLAS, I/F with external (wrt ATLAS) services and sub-systems. A table summarises the data (type of data, volumes and rates, where possible) which flow in/out of tdaq.

Strong link of this section to Chapter 2, "Parameters".

Should this section move to Chapter 12, "Interfaces"?

5.2.1.1 TDAQ interfaces to ATLAS

Indicate what are the interfaces between TDAQ and ATLAS and reference where they are documented.

5.2.1.2 External interfaces

Given the external (i.e. non ATLAS) system and services as highlighted previously, define and reference the interfaces (and indicate the responsibilities).

Finally a summary table, which for any given interface, as defined in the previous section, say what type of data (viz. raw data) is exchanged. Possibly make references to data volumes/rates (where applicable) as documented in Chapter 2, "Parameters".

5.3 TDAQ decomposition

The purpose of the section is to show how TDAQ is organised (the system as such, not necessarily managerially) internally. The internal organisation is looked at from three perspectives: what function are performed by TDAQ, how functions are associated to TDAQ blocks, and a very abstract categorisation of internal elements. Generality (as opposed to implementation) and complementarity of views is stressed.

5.3.1 Functional decomposition

As it was done in the overview chapter, indicate what functions TDAQ is required to provide: detector R/O, event transport, event selection, detector control, monitoring, operation, etc.

This may be a repetition but I suggest it is a useful repetition.

Functions get then assigned to TDAQ building blocks: ROS, LVL2, ROI mechanism, EB, EF, controls, DCS, etc. This section defines what the building blocks are(functions provided and boundaries).

The purpose of the second part of the section is to introduce the concepts of: ROS, LVL2, ROI collection, Event builder, event filter, DCS, tdaq control, etc. As mentioned in the introduction DCS is handled in this chapter as a black box with its own function.

5.3.2 Component categories

Here we characterise the components of TDAQ in terms of broads categories of elements: buffers, processors, supervisors and networks. It will be indicated what buffers (decouple parts of TDAQ, smooth differences in performnces between parts of TDAQ), processors (selection at

HLT level, monitoring, control), supervisors (L2SV, DFM) to control the flow of the data) and networks (transport the data) do in the system.

Q. is it useful to talk of components in such a broad way? In my opinion yes as long as what is stressed is commonality (viz. a ROB, a ROS, an SFI and an SFO do have very strong common aspects, being all buffers decoupling different functions in the system) and not specific characteristics of a particular component.

5.4 TDAQ generic architecture

Now we put together and refine what we have said in Section 5.3, "TDAQ decomposition". The generic architecture is built upon and justified on the basis of that section. A backup document may be needed if more detail is required.

5.4.1 Architectural components

This is the list of the components which are visible at the level of the architecture, there is no a priori problem with the terminology, however the list should include what is relevant in terms of functions, building blocks and abstract elements.

Should include the functions and components identified in Section 5.3, "TDAQ decomposition": ROD, RRC, ..., RoIB, L2SV, online software major components such as control, DB, information & message services, monitoring. For each component the following information should be provided: a definition or purpose (i.e. its function), and required performance. This section is neutral with respect to possible implementations.

Why generic (and “unorthodox”) names such as RRC and RRM, ROB instead of ROBin? The intent is to indicate that at that point in the system something is needed with a certain functionality (to connect and possibly mpx ROLs to ROBs, etc.). I do not mind using different termonology, provided this does not suggest either a definite option (e.g. a point to point link, or we use the same network for lvl2 and EB) or a specific implementation. An S-Link implementation (for example), or the ROBin, are instead object for the baseline section.

5.4.2 Generic diagram

Something like what is shown in Figure 5-1 (to be discussed). The diagram is intended to capture 1) the way TDAQ has been organised and is organised today (i.e. it builds upon tdaq history) while staying generic as far as options (such as how do we go from ROD to ROB, how do we group, if at all, ROBs, how the networks are implemented) are concerned. Options and implementation get spelled out in the section concerning the baseline. The fact that there are separate blobs for EB and LVL2 reflects the fact that there are two separate functions; this does not exclude that the baseline may merge the two functions onto a single network.

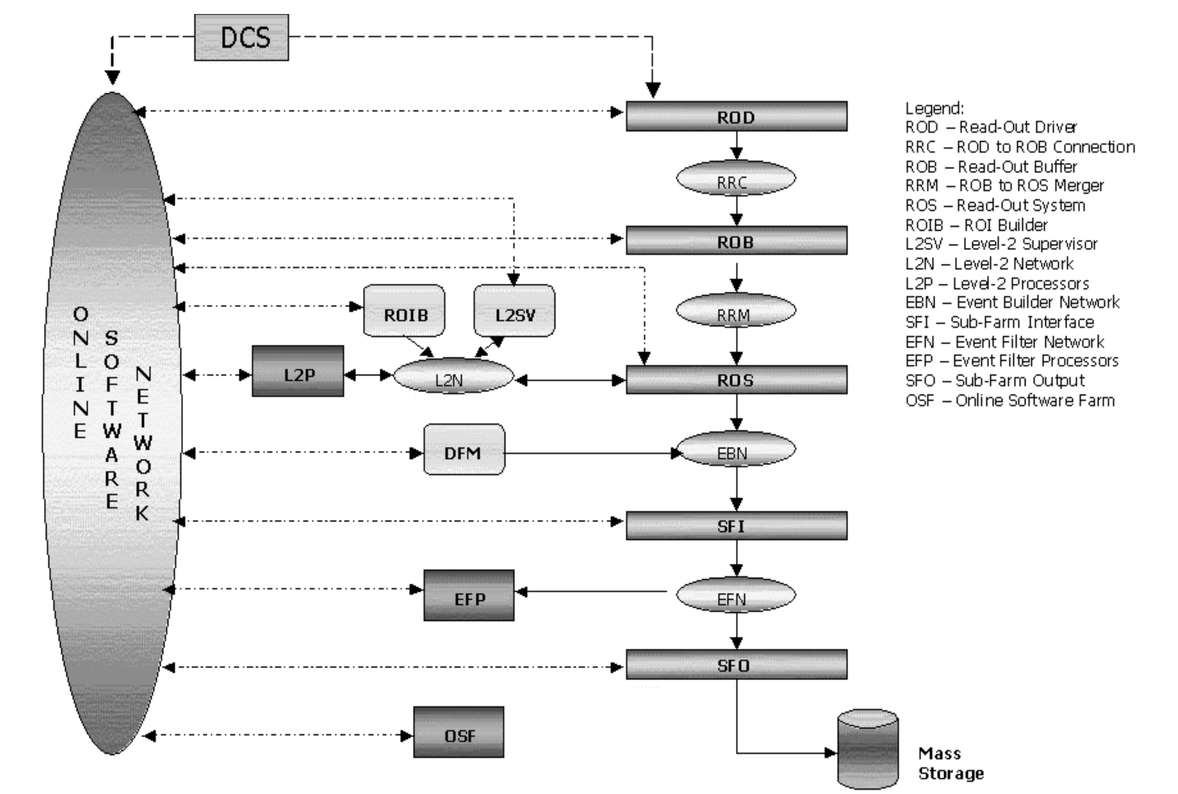


Figure 5-1 Generic diagram.

There are missing things? yes, I suggest it is up to the “views” below to e.g. go into more details as regards their (the view’s) parts. For example a ROB has a data flow view as buffer, but also a control view, a data base view.

The drawing has to be corrected to show LVL1 and its connection to the ROIB, the relationship L2SV/

DFM. Which one can do after we agreed upon the principle

5.5 TDAQ data flow architectural view

Specialise generic architecture for the purpose of Data flow.

Shall contain: functional decomposition into DF packages and sub-packages; interfaces and boundaries between DF packages and sub-packages; main use-cases realisation; "Event control and event flow" view which will include the rates and data volumes between DF packages and sub-packages (including type of communication).

5.6 TDAQ controls and supervision view

Specialised generic architecture for the purpose of control and supervision (eg show local controllers).

Includes both DCS and Online controls.

5.7 Information sharing services view

Specialise the generic architecture for the purpose of information sharing services provided by the online software.

5.8 TDAQ data base view

Data base architecture: including where access to (in and out of) databases is done.

5.9 HLT view

The HLT issues relative to the generic architecture. It should probably include the organisation of the EF and LVL2 blobs, how HLT gets at the data.

5.10 How partitioning is realised on the architecture

The definition of partitions and the allowed operations are defined in Chapter 3, "Operationial requirements for the TDAQ system". Here we make reference to that and we show how the generic architecture defined can be partitioned. What is needed, what is affected, what can be done (and what cannot be done)

5.11 Scalability of the system

How the generic arch. can scale in performance (probably wrt to L1rate). Viz. what has to be expanded, and how. A strategy which show that the architecture can scale.

5.12 Baseline architecture implementation

The baseline would be defined in terms of concrete implementation of the generic components defined above (e.g. what is used to connect ROD to ROBs, the rob is a ROBIN, etc).It certainly should spell out what the generic components are (for example we have a bus based ROS, we use a point to point link between for the ROL) and probably suggest a physical implementation (e.g. this switch is gigabit ethernet with this size). We could also have a sub-section which indicates options (possibly a small number) which one might wish to consider later on.

The justification of the validity of the overall architecture is to be spelled out in Part 3 System Performance.

5.13 References

5-1 Document from Architecture working group on global architecture.

5-2 DataFlow Architecture document.

5-3 ROS Architecture document.

5-4 Data Collection Architecture document.

6 Fault tolerance and error handling

6.1 Fault Tolerance and Error Handling Strategy

General principles:

- robustness
- priority to system efficiency and data quality
- minimization of system down-time
- minimization of single point of failures -> redundancy where affordable,
- failing components must affect as little as possible the functioning of other components
- standardization of error identification
- hierarchical supervision but de-centralized error handling and recovery
- automatic recording of every each failure situation

6.2 Error Reporting Mechanisms

- local fault detection if possible
- error identification
- classification of errors
- error reporting and distribution to clients via a TDAQ central reporting system

6.3 Error Recovery Mechanisms

- central supervision but de-centralized actions by local or regional error recovery
- standardized and/or categorized action on fault depending on the nature of the fault and the DAQ status

6.4 Fault Tolerance

- provide customizable framework for diagnostic and expert system
- emphasis on system robustness
- aim for self stabilizing behaviour in components and regions
- automatic error response and recovery (expert system)
- provide suite of test and diagnosis sequences for each system component
- start acquiring expert knowledge already during the development phase

- identify errors affecting the data quality and mark this in the event
- error prevention due to standardization, redundancy testing, failure recording and analysis

All components interact via well-defined interfaces with other components. If a component does not react properly (e.g. provide responses to requests, be it via the network, a bus etc) this should be considered a fault and the reaction of the component as well as the implications e.g. for the event data should be described.

6.5 Requirements on Components

- minimize the number of single point of failures
- redundancy where affordable
- standardize on error categories
- each component has to provide its test and diagnostic
- an error in the component must affect as little as possible the correct functioning of other components
- provide clear and unique error messages per fault
- verbose logging on actions available on demand

6.6 Typical Use Cases

Describe how a component would react to some typical faults both in a global approach and discussing any system specifics.

ROL (flow control, missing ROD fragments, failure); DF applications (failure of one or more); control and / or event data messages (packet loss, flow control, QOS (peer to peer or switches). Results from modelling may be used to justify.

6.6.1 Non Critical Items

Should include how we deal with a dead ROB.

6.6.2 Critical Items

Should identify which ones are single points of failure, what fault tolerance to build in and how to implement fault tolerance.

6.6.3 Reliability and fault tolerance in the Data Flow

This section was moved from Chapter 8, "Data-flow" and inserted as is here. It is most likely that this will be reworked into Section 6.6, "Typical Use Cases".

This section presents the major error use cases of the DataFlow. As a guideline to identifying the error use cases, major should be interpreted as referring to those that have directly influenced the design of the DataFlow. Each use-case is described as having transient, accumulative or persistent effect on the behaviour of the DataFlow. The handling of each use-case is presented based on results of real life tests. The exact layout of this chapter is subject to the identification of the major error use cases.

Each subsection groups related error use cases.

6.6.3.1 Detector read-out

Possible error use cases here are: ROL failure; assertion of one or more of the error bits in the S_LINK end of frame control word, i.e. ROD fragment corruption; assertion of S-LINK LDOWN; missing or out of sequence ROD fragments.

6.6.3.2 Level 1 to RoI builder

This sub section should be a summary of what is detailed in [8-11].

6.6.3.3 Control and event data messages

How the system handles the loss of each type of control message and event fragments separately.

6.6.3.4 Applications

How the system handles the failure of one or more of the applications.

6.7 References

7 Monitoring

7.1 Overview

Necessity to have a good monitoring for

- the TDAQ system
- the various sub-detectors

7.2 Monitoring sources

which ones of the ATLAS systems and sub-systems need to be monitored during data taking

7.2.1 DAQ monitoring

7.2.1.1 Front-end and ROD monitoring

sub-detector front end electronics specific monitoring

- data integrity monitoring
- operational monitoring (throughput and similar, scalers histograms)
- hardware

7.2.1.2 Data Collection monitoring

DAQ specific monitoring

- data integrity monitoring
- operational monitoring (throughput and similar, scalers histograms)
- hardware

7.2.2 Trigger monitoring

7.2.2.1 Trigger decision

simulate the decision of the trigger stages to confirm the quality of the decision

7.2.2.1.1 LVL1 decision

7.2.2.1.2 LVL2 decision

7.2.2.1.3 EF decision

7.2.2.1.4 Classification monitoring

7.2.2.2 Physics monitoring

“Quality” of the physics which is sent to permanent storage

- Rates of different physics channels

7.2.2.3 Operational monitoring

Everything related to the “system” aspects, e.g. transportation of the events or event fragments, usage of computing resources, etc...

7.2.2.3.1 LVL1 operational monitoring

7.2.2.3.2 LVL2 operational monitoring

7.2.2.3.3 EF operational monitoring

7.2.2.3.4 PESA SW operational monitoring

7.2.3 Detector monitoring

Everything which can be derived from the event data concerning the detector operations during the data taking periods

Connections with DCS should be clarified

7.3 Monitoring destinations and means

Where and how (which tools) to perform monitoring operations

7.3.1 Online Software services

7.3.2 Monitoring in the Event Filter

7.4 Monitoring requirements on networks

Monitoring matrix

7.5 References

7-1

7-2

Part 2

System Components

8 Data-flow

8.1 (Possible introduction)

8.2 Detector read-out and event fragment buffering

8.2.1 Read-out link

8.2.2 Read-out subsystem

8.2.2.1 High Level Design

Based on [8-4]. Present the bus verses switch based ROS issues.

8.2.2.2 Design of the ROBIN

Based on [8-5].

8.2.2.3 Implementation and performance

Summarise the performance results based on the prototype implementation.

8.2.2.4 pROS

Main desription here. Short addition in Chapter 9, "High-level trigger".

8.2.3 ROD crate data acquisition

8.2.3.1 High Level design

Based on [8-7].

8.2.3.2 Implementation

8.3 Boundary and interface to the level 1 trigger

8.3.1 Description

This sub section should be a summary of what is detailed in [8-11].

8.3.2 Region of interest builder

There is overlap with Chapter 9, "High-level trigger" on this component and only one chpater should describe it in detail with the other just mentioning the specifics for that chapter.

8.3.2.1 Detailed design

This sub section should expand on the High level design described in Section 5.5. It should be a summary of what is detailed in [8-3].

8.3.2.2 Performance

Based on results with (12 U) prototype, including results of integration studies with Level 1.

8.4 Control and flow of event data to high level triggers

8.4.1 Message passing

8.4.1.1 Control and event data messages

Introduce the types of messages, the flow of messages, message rates and the bandwidths required. Concluding with the choice of link technology.

8.4.1.2 Ethernet

This section should introduce the key features (i.e. VLANS, QoS, switches, flow control) supporting its selection and how they will be used. Should also summarise, based on [8-8], the basic message passing capabilities in terms of achieved rates, overheads and CPU loads.

8.4.1.3 Design of the message passing component

Presents the main features of the design (high Level enough?) based on [8-9].

8.4.1.4 Performance of the message passing

Presents, based on [8-8], the performance of the message passing component in terms of achieved rates, overheads and CPU loads.

8.4.2 Data collection

8.4.2.1 General overview

This section describes the common model to collecting data for level 2 processing and event building.

8.4.2.2 RoI data collection

8.4.2.2.1 Design

This section should describe the interaction between applications which results in the collection of data at the level 2 processing unit.

8.4.2.2.2 Performance

8.4.2.3 Event Building

8.4.2.3.1 Design

This section should describe the interaction between applications which results in the collection of event fragments to form a complete event at the SFI. Should also include the aspects related to traffic shaping.

8.4.2.3.2 Performance

8.5 Reliability and fault tolerance: this section has been moved to Chapter 6, "Fault tolerance and error handling"

8.5 Configuration, control and operational monitoring

The sub-sections in this section have been moved to other chapters.

Local Controller: moved to Chapter 13, "Experiment control".

Configuration data: moved to Section 10.2, "Databases".

Operational monitoring: moved to Chapter 7, "Monitoring"

8.5 Scalability

8.5.1 Detector read-out channels

This section describes quantitatively how the physical size, performance and control and configuration of the system scales with the "amount" of detector to be read-out.

8.5.1.1 Control and flow of event data

How the number of applications, messages and data volume changes.

8.5.1.2 Configuration and control

Amount of configuration data a function of the amount of detector.

8.5.2 Level 1 rate

How the system performance and physical size scales with respect to the level 1 rate.

8.6 References

8-1 ROS URD

8-2 Data Collection URD

8-3 RoI Builder URD

8-4 Read out system high level design

8-5 ROBIN design documents

8-6 Read sub system test report

8-7 ROD crate DAQ design

8-8 Results of basic comms tests

8-9 Design of the message passing component

8-10 Documents supporting technology choices

8-11 Level 1 - Level 2 interface document

8-12 ROS Local Controller

8-13 DataCollection Local Controller

9 High-level trigger

Chapter 8, Chapter 9, Chapter 10 and Chapter 11 should contain the major components as identified by the architecture.

Details should be provided on design, implementation and supporting measurements. For each component describe: the purpose/function/scope of the component, the performance requirements of the component, the architecture of the component, a proposed implementation, and performance and validation measurements.

The commonalities and differences between LVL2 and EF should clearly be shown.

Detailed design, and performance (where appropriate - performance of some components only relevant/directly measurable as a set-of components) of each component should be described in the sections below.

9.1 HLT Overview

9.2 Level 2

9.2.1 Overview

Includes use RoI mechanism (*i.e. selective Read-out*), requirements and interplay between components.

9.2.2 RoI Builder

There is overlap with Chapter 8, "Data-flow" on this component and only one chapter should describe it in detail with the other just mentioning the specifics for that chapter.

The following comment gives the reason why HLT think the RoIB should primarily be described here: 'Because the function of the RoIB is essentially to combine RoI data from LVL1 and first part of LVL2 processor scheduling. Not clear to us what needs documenting in Chapter 8, "Data-flow"'.

9.2.3 LVL2 supervisor

Suggest main desription is here - as function mainly to control Event flow in LVL2 farm. Clearly needs to have reference back to Chapter 8, "Data-flow" for DataCollection framework. Also needs some description in Chapter 8 for the role in DataFlow - especially the communication with DFM.

9.2.4 pROS

Main desription is in Chapter 8, "Data-flow". Here just a brief note to describe the function from an HLT perspective. i.e. The mechanism to receive the LVL2 result for inclusion in the built event.

9.2.5 LVL2 Processors

Main desription is here. Clearly needs to have reference back to Chapter 8, "Data-flow" for DataCollection framework.

9.2.5.1 L2PU

Description of how the DataCollection framework is used in this case. Includes design and performance in terms of the pure DataCollection tests

9.2.5.2 PSC (PESA Software Controller)

Decscription of the design and implementation of the PSC, how it sits inside the L2PU (including receipt of LVL1_Result and return of LVL2_Result) how it provides various an ATHENA like environment for the Event Selection code and the mechanisms for the algorithms to be configured. Should include some performance results when sitting inside L2PU without running algorithms

9.2.5.3 Data access i/f's

Description of how the PESA DataManager can access RoI data from the L2PU. Not clear that there are meaningful separate performance measurements for this.

Issue - Should we include possible use of FPGA's here or under system performance aspects?

9.3 Event filter

Description of EF DataFlow - event distribution, use of PT's (providing ATHENA environment), use for Event Selection, for Calibration and data monitoring, generation of EF_Result, appending EF_Result to the built event, passing accepted events back to main DataFlow.

9.4 Event selection software

ESS Architecture - requirements, design and implementation - including the main internal components

Configuration, control, supervision and operational monitoring: moved to several other chapters.

Describe here the HLT specific items and issues.

Thus:

1) LVL2 use same LocalController as DataCollection. Need to add something about the operational supervision of the LVL2 processors.

2) EF has the EF Supervision sub-system

3) Concept of sub-farms - how they are defined, and application configured

As far as possible common issues should be described in the OnLineSW chapter.

Issue - Should we include management of the sub-farm fabrics here? (Presumably yes)

Issue - Where do we cover algorithm configuration

(I assume that data and algorithm related monitoring is included in the monitoring chapter).

9.5 References

9-1 LVL2 URD

9-2 EF DataFlow URD

9-3 EF Supervision URD

9-4 ESS Requirements Doc

9-5 ESS Design Doc

10 Online software

An overview of the functionality of the online software if or as far as it is not yet done in Part 1 is given. The scope of the Online software components is defined as TDAQ and Detectors.

10.0.1 The Architectural Model

The hierarchical component model of the online software system will be presented and the composition of the online sub-system with the logical view (diagram) with clear separation between components and well defined boundaries. TDAQ control and supervision, Databases and Information Sharing are introduced and the model of services is explained.

10.1 Control and supervision

10.1.1 Functionality of the control and supervision

The control and supervision package encompasses software components responsible for the control of other TDAQ systems and detectors. It includes facilities for run command distribution and synchronization between the systems, TDAQ initialization and shutdown and run supervision.

The Verification sub-system is responsible to verify the functionality of any subset of the current TDAQ configuration. It uses developers knowledge to organize tests in sequences, analyze test results, diagnose problems and provide conclusions about the functional state of TDAQ components. An expert system is a possible design choice for knowledge representation and reasoning.

The Process Management provides basic process management functionality in a distributed environment. This functionality includes starting, stopping and monitoring processes on different TDAQ hosts.

The Resource Management is concerned with the allocation of software and hardware resources between running partitions. It prevents the operator from trying to perform operations on certain resources with conflicting interests.

The Access Management is a general Online software safety service, responsible for TDAQ users authentication and implementation of an access policy to prevent non-authorised users to corrupt TDAQ functionality.

The User Interface (UI) provides an integrated view of the TDAQ system to the operator and should be the main interaction point. It is foreseen to provide a flexible and extensible UI that can accommodate parts implemented by the detectors or other TDAQ system users.

10.1.2 Performance and Scalability Requirements on the control and supervision

The current understanding of the performance and scalability requirements as requested by the online system users will be listed.

10.1.3 Architecture of control and supervision

remark: the content of this paragraphs may be moved to Section 10.1.1, "Functionality of the control and supervision".

10.1.3.1 Interaction of the control and supervision system with other Online SW sub-systems

Databases, Information Sharing Sequence diagrams, activity diagrams

10.1.3.2 Supervision and Verification

logical view on design

10.1.3.3 Process, Access and Resource Management systems

logical view on design

10.1.4 Prototype evaluation

description of current implementation

performance and scallability of current implementation

might include future technology evaluations

10.2 Databases

10.2.1 Functionality of the Databases

The database package is dealing with persistent information to be shared between the TDAQ system and its applications. It includes the configuration databases describing the system, the online bookkeeper supporting log of run conditions and the interface to the off-line conditions databases.

The Configuration Databases component describes several configurations prepared for different types of runs, allows this description to be available for all TDAQ applications and allows to modify such a description by authorized experts and their processes.

The Online Book-Keeper (OBK) is the system responsible for the online storage of relevant operational monitoring and configuration data. The OBK organizes the stored data on a per-run basis. In order for that data to be useful while doing off-line reconstruction or analysis, the system also provides querying facilities.

The online Conditions Databases Interface (CondDBI) provides an API for accessing conditions data for all TDAQ and detectors processes and to provide preferment access to the conditions-data.

The Conditions Databases are provided by the off-line software group. The online environment is both the provider and the client of the conditions data.

remark: details of the functionality of the Conditions Database interface are till under discussion and the therefore the content of the explication given above may change.

10.2.2 Performance and Scalability Requirements on the Databases

The current understanding of the performance and scalability requirements as requested by the online system users will be listed.

10.2.3 Architecture of Databases

remark: the content of this paragraph may be moved to Section 10.2.1, "Functionality of the Databases".

logical view on design

10.2.3.1 Configuration databases (ConfDB):

ConfDB Data Access library (DAL), ConfDB DAL Generator, ConfDB Monitor, ConfDB repository, ConfDB User Interface

logical view on design

10.2.3.2 Online bookkeeper (OBK):

OBK log info provider, OBK repository, OBK browser, OBK Administrator

logical view on design

10.2.3.3 Conditions database interface

details in preparation

10.2.4 Application of databases to the TDAQ sub-systems

Usage of the databases by the other TDAQ systems, concentrating on differences with general use.

Should be provided by TDAQ systems

10.2.5 Prototype evaluation

description of current implementation

performance and scallability of current implementation

might include future technology evaluations

10.3 Information Sharing

The choice of name for this section is not final. A possible alternative could be "Monitoring services". This would then be appied to the whole of the document where one talks about these services.

10.3.1 Functionality of the Information Sharing Services

The Information Sharing package is responsible for providing a means for sharing and monitoring operational and event data between information providers and information consumers in the TDAQ system. Information is shared between information providers and information consumers in the TDAQ system. The sharing mechanism depends on the type of information. Services are provided for sharing generic information items, errors, events and histograms.

The Information Service (IS) allows to exchange user-defined information between applications. The Error Reporting Service (ERS) provides transportation of the error messages from the applications which detect these errors to the applications which are responsible for their handling.

The Online Histogramming Service (OHS) allows applications to exchange histograms. The OHS is very similar to the Information Service. The difference is that the information which is transported from the providers to the receivers has pre-defined format.

The Event Monitoring Service (EMS) is responsible for transportation of physical events or event fragments sampled from well-defined points in the data flow chain to the applications which can analyse them in order to monitor the state of the data acquisition and the quality of physics data in the experiment.

10.3.2 Performance and scalability requirements on Information Sharing

The current understanding of the performance and scalability requirements as requested by the online system users will be listed.

10.3.3 Architecture of Information Sharing services

remark: the content of this paragraph may be moved to Section 10.3.1, "Functionality of the Information Sharing Services".

logical view on design

10.3.3.1 Information Service (IS)

InfoProvider, InformationService, InfoConsumer

logical view on design

10.3.3.2 Error Reporting Service (ERS)

InfoProvider, Error Reporting Service, ErrorReceiver

logical view on design

10.3.3.3 Online Histogramming Service (OHS)

HistoProvider, HistogrammingService, HistoReceiver, Histogram Display

logical view on design

10.3.3.4 Event Monitoring Service (EMS)

Event Sampler, Event Monitoring Service, EventReceiver, EventDump

logical view on design

10.3.3.5 Application of information services to the TDAQ sub-systems

Usage of the information services by the other TDAQ systems, concentrating on differences with general use.

Should be provided by TDAQ systems

This sub-section should only exist if the information is not already covered in Chapter 7, "Monitoring".

10.3.4 Prototype evaluation

description of current implementation

performance and scallability of current implementation

might include future technology evaluations

10.4 References

10-1

10-2

11 DCS

Chapter 8, Chapter 9, Chapter 10 and Chapter 11 should contain the major components as identified by the architecture.

Details should be provided on design, implementation and supporting measurements. For each component describe: the purpose/function/scope of the component, the performance requirements of the component, the architecture of the component, a proposed implementation, and performance and validation measurements.

11.1 Introduction (Move to Section 1.2.6?)

Scope, mandate and definition of the DCS

Defines what is covered by the DCS and what is excluded.

11.2 Logical Structure (Move to Section 5.6?)

Detector and External Systems

11.3 DCS Architecture (Move partially to Section 5.6?)

This section describes the overall architecture of the DCS and introduces the main compoments, namely the Back-End sytem and the Front-End. It also describes the hierarchical organization of the BE system in three levels:

- Global Control Stations
- Local Control Stations
- Subsystems Control Stations

It also places the DCS in the context of the general TDAQ system. The DCS is seen a self-contained system. The interfaces of the DCS with other parts of TDAQ are described.

-------------------------------This could be the beginning of Chapter 11-------------------------------------

11.4 DCS Components (Fictitious section, could be Chapter 11)

This section describes the two main components of the DCS, namely the Front-End and the Back-End system.

11.4.1 The Back-End System

11.4.1.1 Organization of the BE system

This sub-section moves to Chapter 13, "Experiment control".

It describes the functions of the three levels of the Back-End sytems:

- Global Control Station
- Local Control Stations
- Subsystems Control Station

11.4.1.2 Software components

- PVSS
- PVSS Framework
- Other software components (data and alarm displays, web services, root interface, etc.)

11.4.2 Front-End System

11.4.2.1 Embedded Local Monitor Board

- Features (Description of HW and SW of the ELMB)
- Performance (Accuracy, stability, etc.)
- Radiation Qualification
- Magnetic testing
- Add ons (Interlock box, motherboard, DAC, etc.)
- Canbus Topology
  - Bus Powering
  - Bus and Node Supervision

11.4.2.2 Other standard FE equipment (HV, LV)

VME, PLC, etc...

11.4.2.3 Connection to subdetector specific equipment

Muon alignment system,LAr purity control system, CS calibration source of the Tilecal, etc.

11.4.3 Read-out chain

Elements of the readout chain

Comprising ELMB, Kvaser interface card, CANopen OPC server and PVSS. Here is where the CANopen OPC server should be described

11.4.3.1 Work load distribution

This section describes the functions performed by the different elements of the readout chain

11.4.3.2 Performance

Here is where the results of the ELMB Full Branch test must be included

Something about the operation in TCC2

11.4.3.3 Scalability

A few comments on the outcome of the results presented in the previous section

11.4.4 Physical distribution of the components

11.4.5 Applications

Covers: Common Infrastructure controls (racks, cooling, radiation monitoring, etc.)

-------------------This could be the end of Chapter 11---

11.5 DCS Data (Split and move to Section 1.3 and Section 3.7 on Databases?)

11.5.1 Configuration data

Data Volumes and rates

Databases?

11.5.2 Conditions data / Output data / Measured data?

Data Volumes and rates

Databases?

11.6 Connection to DAQ (goes to Chapter 13)

Goes to Chapter 13, "Experiment control".

11.6.1 Functional Requirements (???)

11.6.2 Connection points (???)

11.6.3 DAQ-DCS Communication software (Move to Section 12.2?)

11.7 Operation (Move to Chapter 3)

In Physics, Global, subdetector, subsystem modes.

11.8 External Systems (Move to Section 12.1?)

11.8.1 LHC

The LHC will be interfaced by means of the Data Interchange protocol to be provided by JCOP

11.8.2 CERN services

The CERN Services will be interfaced by means of the Data Interchange protocol to be provided by JCOP

11.8.3 Detector Safety System

11.8.4 Magnet

11.9 Organization and resources (Goes to Part 4)

This section should be moved to Chapter 18.

11.10 Work Plan (Goes to Part 4)

This section should be moved to Chapter 19.

11.11 References

11-1

11-2

11-3

12 Interfaces

12.1 External to TDAQ

12.1.1 LHC machine

12.1.2 Detectors

12.1.3 Off-line

12.2 Internal to TDAQ

12.2.1 LVL1

12.2.2 ...

12.3 References

12-1

12-2

13 Experiment control

This chapter has been positioned after all other chapters in this part because it relies on information presented in the previous chapters.

13.1 Introduction

This chapter brings all the control elements together to show the overall control mechanism/ strategy. The concepts presented here will have already been introduced in Chapter 5, "Architecture".

13.2 Control coordination

Explain the three different finite state machines present in the system, namely online, DCS, and machine, and their synchronisation.

13.3 Control scenarios

Control scenarios of:

- different types of calibration.
- physics run.
- operation outside a run.

13.4 Sub-system control

The information presented here might already be presented elsewhere in which case this section is not needed.

detector control

HLT farm supervision

DF control

online software concepts

13.5 References

13-1

13-2

Part 3

System Performance

14 Physics selection and HLT performance

14.1 Introduction

Recall the strategy (as in Section 4) and the inclusive approach (more details later on non-inclusive selections).

Explain the use of selection algorithms at different levels and the selection sequence

Highlight the use of updated detector geometry (also in start-up phase, *i.e.* staged, implementation) and (wherever possible!) the use of realistic data and communication schemes, the use of fully simulated data with proper pile-up

2x10^{33} cm^{-2}s−1 (we need a FM variable here!) and $L = 1.0 \times 10^{34}$ $\text{cm}^{-2}\text{s}^{-1}$ different approaches

Find a clever way to explain how we will do bricolage when we cannot use the full-fledged schema to get byte stream, decode, apply HLT algorithms, derive features, take decision.

Don't do like CMS (because we can) and do not explain all the things that we already said in Physics TDR and HLT TP, but only reference them.

14.2 Common tools for selection

Describe the tools (algorithms) used at the different levels, with a focus on the LVL2 detector reconstruction (e.g. Calorimeter clustering, ID tracking, etc).

Highlight the approach of the Algorithm Task Force, the use of common tools for different selections. Do not forget that EF is "inherited" from off-line and explain how much of the full analysis chain is retained here.

Link also with description of PESA-SW, Steering, Data Access, etc.

14.3 Signatures, rates and efficiencies

Derive from Trigger Menus (of Section 4) list of representative physics signatures (*à la TP*).

14.3.1 e/gamma

Emphasis on this selection: most of explanations will be here.

14.3.2 Muon selection

Differences wrt TP, low-p_T signatures (see later for B), barrel approach, end-cap ?

14.3.3 Tau/jets/$E_T$miss

Highlight major discovery channels, for taus probably start using bricolage

14.3.4 b-tagging

Define on-line strategy for this, explain why we think we need it, discuss implications for jets thresholds (and hence rates)

14.3.5 B-physics

Agree on ATLAS policy. Explain strategy, start with di-muons selection, fill-in with other low-p_T signatures with decreasing instantaneous luminosity (some of this probably already in Section 4)

For each of the above, go through the list of signatures and derive numbers for rates and efficiencies (both HLT wrt LVL1 and between LVL2 and EF steps)

See also recent Saul's comments to include test-bed results in performance evaluation

14.4 Event rates and size to off-line

Define present ideas about data compression and reduction, zero suppression for LAr (and TRT?): this might be probably be elsewhere as well. Differences between zeros at the EF and loss-less data compression in the ROSes.

Global table on rates for initial and high luminosity, implication for off-line reconstruction (costing, later)

14.5 Start-up scenario

Should be here? Picture a global approach on how we are going to handle, at the selection level, the first year of running, assuming a certain machine scenario. It is probably very appealing for LHCC

14.6 References

14-1 *ATLAS detector and physics performance technical design report*, CERN–LHCC / 99–14 / 15 (1999)

15 Overall system performance and validation

15.1 Introduction

- Definition of validation of rate capability, its context and scope.

- Summary of validation process

15.2 Integrated Prototype

Description of the integrated 10% system

15.2.1 Laboratory setup

- machines, networks, OS platform(s), hardware emulators (if any)

- refer to architecture and components chapters for details

15.2.2 Description of the measurements

- scope of the measurement (what parameter(s) of chapter 2 are we testing)

- parameter space covered

15.2.3 Results

- prototype results

- comparison with required performance

15.3 Title?

Comparison of requirements for full system from paper model and extrapolation of results of integrated prototype

15.4 Computer model

15.4.1 Methodology

15.4.2 Result of testbed model

15.4.3 Results of extrapolation of testbed model and identification of problem areas

15.5 Title?

15.5.1 Technology tracking up to LHC turn-on

15.5.1.1 Network technology

15.5.1.2 Processors

15.5.2 Survey of non-ATLAS solutions

(a reality-check on ATLAS approach?)

15.5.3 Implication of staging scenarios

Re-interpretation of performance numbers for staging scenarios

15.5.4 Areas of concern

15.6 Conclusions

15.7 References

15-1

15-2

Part 4

Organisation and Plan

16 Quality Assurance and Development Process

remark: this chapter also addresses the Quality assurance during deployment. The text only tries to give a first flavour of the items to be included there and others may have to be added. It may have to be discussed amongst the editors if this is the right place to present them or if this point is better covered entirely in chapter 6.

16.1 Quality Assurance of the TDAQ components

Quality assurance during the production of harware and software systems is provided for with the adoption of a development framework for DAQ components. The development framework consists of distinct development phases. At the end of each phase a set of deliverables is provided.

This framework is complemented by guidelines, checklists and standards, internal reviews, templates, development and testing tools and coding standards. Those are being adopted as common working practice and help for error removal and error prevention in the system.

A powerful release management system and a convenient working environment has been put into place.

16.2 The Development Process

The software development process (SDP) in Atlas TDAQ provides the structure and the sequence of activities required for development. A basic framework is provided to guide developers through the steps needed during the development of a component or a system. Continual review and modification of the SDP provides it with the flexibility to adapt to the evolution of the components and systems.

Many of the recommended approaches in the SDP are also applicable to the development of hardware components or sub-systems involving both software and hardware. The SDP consists of the following phases (diagram): Brainstorming, Requirements, Architecture and Design, Implementation, Testing, Maintenance, complemented by reviews. Emphasis on the phases will evolve within time.

During event production, the emphasis will be put on maintenance and regular atomized validation testing

remark: it will depend on the level of required detail if the following items will be split into sub-paragraphs or not.

16.2.1 Requirements

requirements gather process, documentation

16.2.2 Architecture and Design

guidelines on diagrams, common notation

16.2.3 Implementation

coding standards, checking tools

16.2.4 Inspection and Review

organisation, aims, benefits

16.2.5 Component Testing and Integration Testing

organization of tests, testing doc, testing tools, test and diagnostic facilities

16.2.6 Maintenance

16.2.7 Experience/Justification

explain why we think it will work

16.3 The Development Environment

release building and its use, platforms, etc.

16.4 Quality Assurance During Deployment

16.4.1 Quality Assurance of operations during data taking times

The quality of the DAQ system must be assured when itis in use during the setup and installation phase of the Atlas data acquisition together with the detectors. Correct and smooth data taking shall be aimed for during callibration and physics event production.

Fault tolerance build into the system from the start and efficient error handling provide the basis (see chap.6 for details). System redundancy to reduce possible single point of failures is foreseen where affordable (give examples).

16.4.2 Quality Assurance of Event Data during data taking

The integrity of the event data must be assured during data taking sessions. It will be checked at various levels (t.b.defined) in the read-out chain. Incomplete events will be marked as suchdirectly.

16.5 References

17 Costing

17.1 Initial system

17.2 Final system

17.3 Deferral plan

17.4 References

17-1

17-2

18 Organization and resources

Should the geographical, racks, power supplies, and cooling issues be addresses in this chapter or in the system component ones?

18.1 ...

18.2 References

18-1

18-2

19 Work-plan

Post TDR.

19.1 Schedule

19.2 Commissioning

19.2.1 TDAQ

19.2.2 Tools for detectors

19.3 References

19-1

19-2

This document has been prepared with Release 5.5 of the Adobe FrameMaker® Technical Publishing System using the Technical Design Report template prepared by Mario Ruggier of the Information and Programming Techniques Group, ECP Division, CERN, according to requirements from the ATLAS collaboration.

To facilitate multiple author editing and electronic distribution of documents, only widely available fonts have been used. The principal ones are:

Running text:	Palatino 10.5 point on 13 point line spacing
Chapter headings:	Helvetica Bold 18 point
2nd, 3rd and 4th level headings:	Helvetica Bold 14, 12 and 10 point respectively
Figure and table captions:	Helvetica 9 point