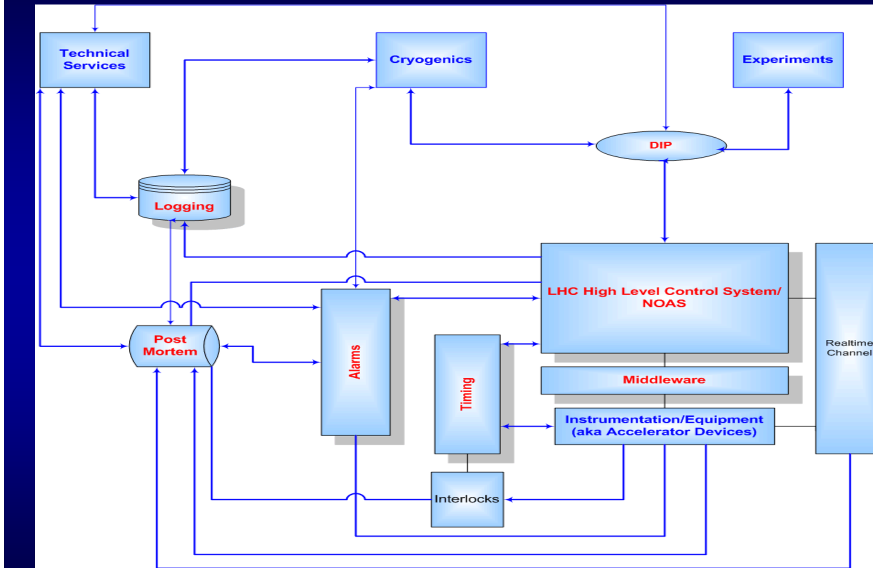
Common Control Facilities

Summary of session 2
LHC-CP workshop June 2003

• Alarms	Mark Tyrrell
• Logging	Maciej Peryt
• Post Mortem	Jörg Wenninger
• DIP	Wayne Salter
• Middleware	Kris Kostro
• OASIS	Javier Serrano
• Architecture	Eugenia Hatziangeli
• Timing	Julian Lewis
• Naming	Ronny Billen

Context

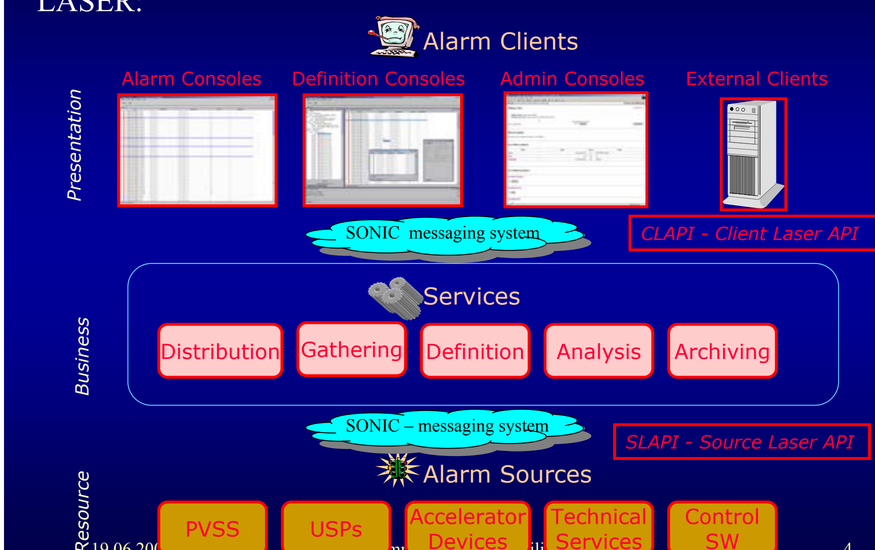


Alarms

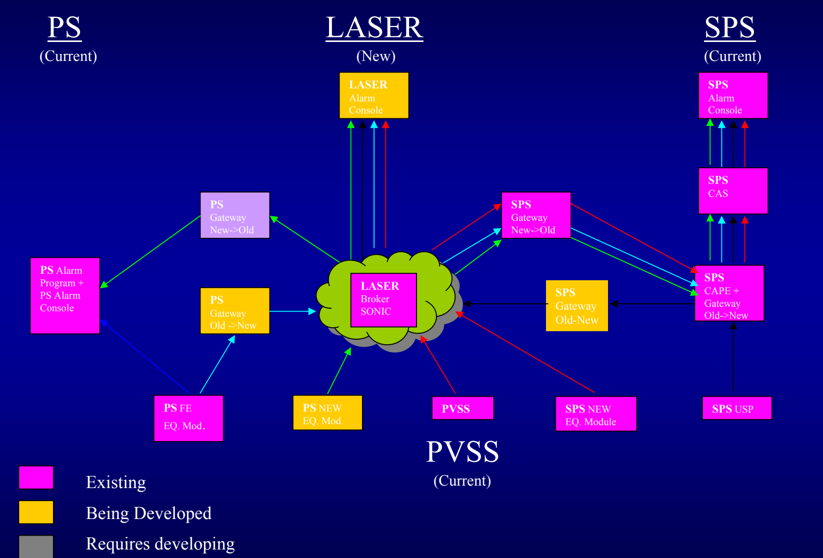
- **Two “current” systems**
  - **The PS alarm system**
  - **The SPS, CERN technical services and safety alarm system**
- **One new system**
  - **LASER: Lhc Alarm SERvice – using the ‘new’ technology**
    - **Source API: ‘C’ or Java, J2EE Application Server – EJB’s, JMS (SONIC messaging system), Client API: Java, NetBeans / using the Gui Platform (GP) wrapper for alarm consoles**
- **Alarm team needs to provide:**
  - **a continuous service to our existing users:**
  - **graceful transitions from ‘current’ to ‘new’.**
- **NB: PARTS OF THE CURRENT SYSTEM WILL REMAIN BEYOND 2004…**

Where are we going ?

LASER:



Where are we going ?



TT40 & QRL

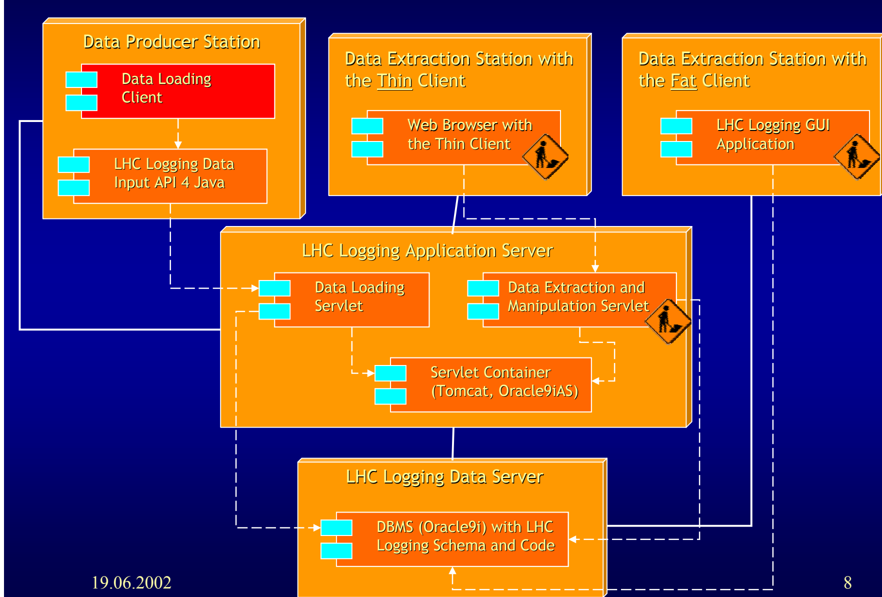
- What we will guarantee for the above:
  - alarm consoles to display alarms from any source:
    - current alarm consoles
    - current alarm archive
    - use the above to test and verify the LASER prototype
- Hope to provide a LASER vertical slice with new alarm consoles and integrated archive
- We will have alarm facilities for TT40 and QRL.
- Questions:
  - FESA?
  - Databases: A mess
  - DIP: Do we need two means of inter-domain connectivity?
  - Alarm Review Process? What is, and is not, an alarm, and its resulting priority must be given serious, and professional consideration.

Logging

Logging

- Analysis, design, procurement of Logging Facilities for future LHC Controls System
- Within the scope is:
  - Analyze experience, capture requirements
  - Implement first version to support QRL
  - Logging data from TT40 extraction tests.
  - Investigate interface with Alarms and Post-Mortem systems
- Objectives
  - Establish logging facility for TT40 and QRL, scalable to LHC
  - Major project review after initial validation

Architecture



Logging

Logging: components

- Data Input API 4 Java
  - Prototype available
- Thin Data Extraction Client
  - Functionality similar to stride – String2 Data Extractor, but different platform (Java).
  - Still looking for charts package to be used both in thin and in fat client.
    - ILOG JViews 5.5 Evaluation
  - Will be available for TT40 Extraction Tests

European Organization for Nuclear Research

CERN **LHC Logging Project**

Sub-project of the LHC Controls Project

[Home | Documents | Software]

· Data Input API

Data Input API

In order to use the API you will need the JAR file containing the classes, source code of the examples and some configuration files. The library depends on Oracle XML Class Generator for Java and Oracle XML Parser for Java v2. If you do not have these products on your computer, you will have to download them as well. You can find all necessary information to start using the API in the javadoc documentation.

- The library itself [download]
- Oracle XML Class Generator for Java [download]
- Oracle XML Parser for Java v2 [download]
- API Documentation [view]

top

Logging

Next step: TT40 Extraction Tests

- API to be tested by SPS 2001.
- We need to test the full chain with real data:
  - Equipment → CMW → Logging → query.
- Logging on cycle basis: higher data rates than initially assumed.
- Still not clear what data types apart from scalar data will be stored.
  - We expect to store vectors of numbers – like profile data.
- If OK for TT40, then QRL should be satisfied too.
  - QRL: waiting for the clients to show up!

TT40: a functionality showcase.

We are (finally) ready to log data.

Post Mortem

Post Mortem

Importance of the post mortem system stressed.

- To understand when, why and how interlocks are triggered.
- To determine the initial cause of a ‘problem’, to adjust interlock thresholds… we must be able to see the last moments before the beam disappears in the dump block !

Post Mortem

Post-mortem ingredients

- Every LHC equipment and diagnostics system must implement a circular PM buffer of appropriate depth holding the latest data (example : last 1000 turns for beam instruments …).
- Data must be time-stamped to ~ ms or μs depending on type.
- The PM buffer must be frozen by an external post-mortem event or by self-triggering.
- The PM data must be combined to form the post-mortem event data: size ~ few Gbytes.
- The PM data must be automatically analyzed. ‘Digested’ information must be generated for operations.
- The PM data must be stored – the most relevant data must be stored for the lifetime of the LHC. Some of it will be important for INB.

PM Data

Post Mortem

- **Data should be self-describing**
- An 'event builder' is required to :
  - assemble the data (push or pull ?).
  - assign it a unique PM event number (key).
  - verify data integrity and completeness.
  - store the data on disk for immediate analysis.
  - possibly send it to long term storage
- **Wading through it to find information of relevance will need to be fast, intelligent, automatic etc.**

➤ A look towards the experiments

- **Huge amounts of data per event and the tools to deal with it**

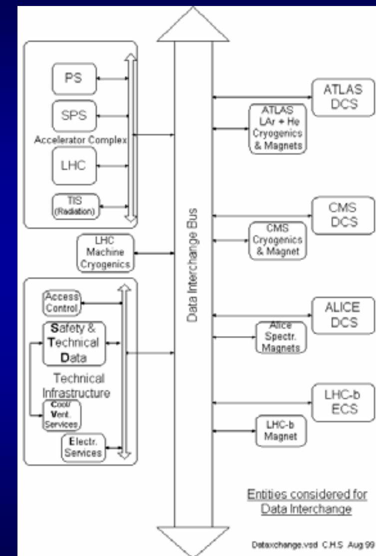
- Milestone # 1 : sector commissioning in 2005 !

HELP NEEDED!!!

DIP

DIP

"DIP should be able to exchange relatively small amounts of real-time data between very loosely couple heterogeneous systems."



Phase II

DIP

- **Review user requirements**
- **Identify system requirements**
- **Review products in use at CERN for applicability**
- **Define the DIP protocol – end of 1st quarter 2003**
- **Select a suitable product**
- **Develop a prototype implementation of DIP API for end of 2nd quarter 2003**

- **User and system requirements discussed at length**
- **5 possible products considered for potential prototyping**
- **2/5 selected for prototyping – some wrangling**

Some DIPPY assumptions

DIP

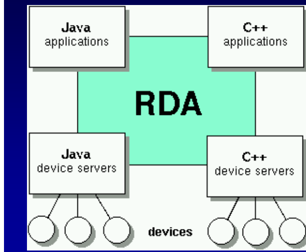
- **On reconnect the client can decide either to get automatically the 'current' value for all the data items she subscribes to or not to get it**
- **On-change or at fixed frequency data exchange**
- **DIP should support arrays but not necessarily more complex structured data.**
- **Self-describing data not necessary**
- **Security**
  - **Simple authentication mechanism**
- **Administration**
  - **Status of system, is a publisher alive etc. etc.**
- **Support**
  - **API**
  - **Servers etc.**

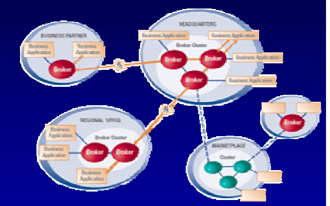
Status

DIP

- SonicMQ selected as first candidate – implementation of DIP API nearing completion and tests expected to start soon
- Choice between Oracle AQ and DIM for 2nd candidate
  - DIM chosen
- Evaluation of second product to be followed by selection of one of the evaluated products
- DIP definition planned for early 3rd quarter 2003
- DIP prototype implementation 3rd quarter 2003 for the QRL tests
- AB-CO to decide whether to use DIP for the alarm system?
  - DIP requirements not equal Alarms requirements

Middleware - Guided tour





- Device/property - get/set pub/sub
- Data container
- Cycle selector
- Acquisition stamping
- Java clients
  - CMW provides the low-level RDA API
  - AB/CO/AP provides the high-level API

- CMW C++ servers
- Naming & configuration
- Admin facilities

Performance looks good

Deployment status

- Access to existing PS & SPS equipment via CMW
  - Deployed in PS since 2002
  - Deployed in SPS as SL-Equip gateway for Excel passerelle and RF, can also be deployed natively on LynxOS F-E
  - Gateway for CESAR (EA renovation) to access all EA equipment
- TT40
  - Generic Biscoto CMW server developed and tested with BCTFI
  - Other Biscoto instruments: BPMI, BTVP
  - Via CMW Equip servers: beam loss (BLR), SPS orbit (MOPOS), other?
  - Currently helping to set up “shot-by-shot” logging using subscription.
  - Beam Interlock System (BIC), currently under development, is using PS-type CMW server.
- LHC
  - Power converters, QPS, and RF development ongoing

CMW Conclusions

- CMW in The AB/CO standard for remote equipment access
- The infrastructure is finalized
- Existing PS/SPS equipment is accessible
- Requirements for TT40 commissioning, HW commissioning, LHC, as far as known, have been addressed. Working closely with equipment groups, CO/AP and operation.
- To do
  - The upcoming equipment server framework has to be integrated.
  - PVSS has to be interfaced with CMW
  - Standardize usage of CMW across various servers
  - Exploit full potential of CMW servers.
  - Develop CMW server for FECOMSA standard front-end server framework

High Level Controls, Architecture & Development

Software Development

- High level development
  - Software Development
    - Unified Software Development Process
  - Software Implementation
    - Java (IDEs etc..)
    - C/C++ (legacy and PVSS interfacing)
    - Extensible Markup Language (XML)
    - PVSS
  - Tools
    - Object/Relational mapping TopLink
    - Jcover, Junit, Together Audit (testing)
    - OptimizeIt (optimisation)
    - JStyle (Quality Analysis)
  - Software building, support by AB/CO/AP
    - “common build”, AP/CO/AP made tool for Java, Based on Ant
  - Software Configuration & Change management
  - Project management
    - Goal Directed Project Management (GDPM)

High Level Controls, Architecture & Development

High level controls

Operational Applications

- Application Standard Components
  - GUI platform
  - Data visualization
  - JAPC
- Application Deployment & Management
  - Java Web Start (JaWS)
  - It is provided by SUN, as part of the JDK
  - Launches Java applications, as a set of jar files, directly from the Web (slwww)
- Applications Management
  - Console Manager (YACoMa)
- Application Server
  - Collaboration underway to provide Oracle 9i Application Server for Platform for Development (May 2003) & Platform for Operational deployment (upcoming)
  - Experience to be gained

High Level Controls, Architecture & Development

Conclusions - Development

- Software Development Process (analysis, implementation, tools, build, version control etc.) is well established and used successfully already in several projects
- Set of recommended tools is available
- 21st technology firmly embraced
- 3-tier has been chosen
- Using J2EE/EJB to implement it (Industry standard, Cool and component based)
  - Several projects (CESAR, LASER, SPS2001) are using 3-tier (J2EE/EJB) architecture => built up confidence
  - New projects (BIC, OASIS) are adopting this architecture
- Challenge: setup/tune the Application Servers to achieve the performance, reliability & availability needed for critical applications (J2EE WG)
- Cesar and TT40 tests will be a validation

High Level Controls, Architecture & Development

Conclusions - Development

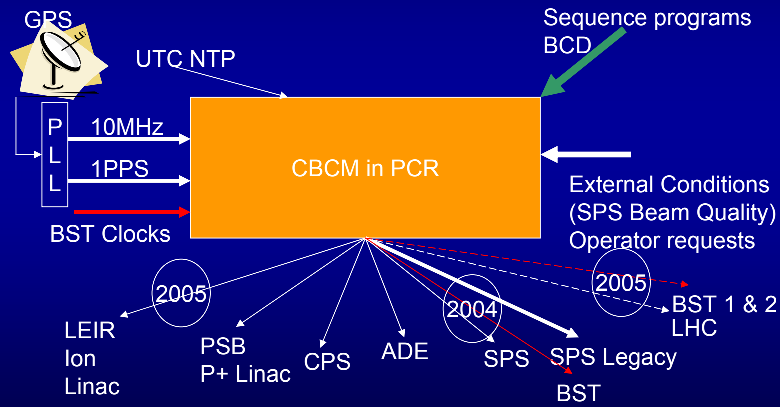
- High level controls components are prepared, based on experience with operating large machines: PS, SPS, Transfer Lines, LEP and the commonalities between them:
  - LHC controls applications software will be based on
    - Software technology choices and standard components made for SPS2001, CESAR, LASER
    - The infrastructure deployed for those projects
    - Aim for common solutions
- A clear AB/CO objective is to reduce diversity in the available CO solutions and deploy common services and components across all accelerators

- Projects progressing well, driven by realistic objectives based on the requirements for TT40 tests and other LHC major milestones
- SPS and TT40 will be used as test beds for our new controls infrastructure and software technology choices
- Need to clarify controls requirements for the next LHC major milestones for 2004 and beyond and start preparing for them

Timing

Timing

All the CERN accelerators will share the same Lego for timing. Generation, Transmission, Reception hardware. The software drivers and high level equipment access is also shared.



LHC

Timing

- LHC GMT driven from CBCM
- LHC BST driven from CBCM
- Same hardware used in PS and SPS
- Same software used in PS and SPS
- Strongly UTC time based
- Telegrams
- Strongly coupled with the SPS during beam transfers

Hardware

Timing

- Lot of development and prototyping
  - CTRA VME Timing receiver
  - CTRP Timing receiver module
  - CTRV, VME version of the CTRP
  - Prototype BST master card based on the same PCB used in the CTGU
  - CTGU
  - CTGSW
  - Keeping Time with UTC CTGSY Card
  - Optical, ECL, TTL fan outs etc. etc.
- Signal delivery – another big job
  - GMT
  - 10MHz
  - LHC Injection & SPS Extraction Pulses

Incoming

Timing

- TT40
  - Cool
- CBCM
  - Big effort
  - Planned for use in SPS 2004 LEIR startup 2005, LHC startup 2006
  - Sequences, Sequence Manager, Sequence Editor,

OASIS

Open Analogue Signals Information System

Main goal: to satisfy the user requirements gathered by the LHC-CP Analogue Signals Working Group

OASIS

- Aim to use 3-tier architecture
- GUI Application: written in Java. Based EdPlot package
- Middle tier: Enterprise Java Beans on a J2EE server machine.
- Front ends:
  - Equipment Modules using CMW for communication.
  - Real-time task written in C++
  - Linux driver written for the CompactPCI multiplexer modules.

Hardware

OASIS

- CompactPCI with Concurrent Technologies CPU running Linux.
- Acqiris DC270 digitizer modules (4 channel, 250 MHz BW, 1Gs/s).
- Pickering 40-745-501 4 to 1 RF multiplexer (1 GHz BW).
- Acqiris CC105 crate (7.5A on 12V supply). We might switch to Wiener in the future.

OASIS: Conclusions

OASIS

- No major technical stumbling blocks ahead.
- FE Software finished.
- GUI well underway.
- Middle tier designed. Implementation will benefit from AB-CO-AP support.

Naming of parts

NAMING

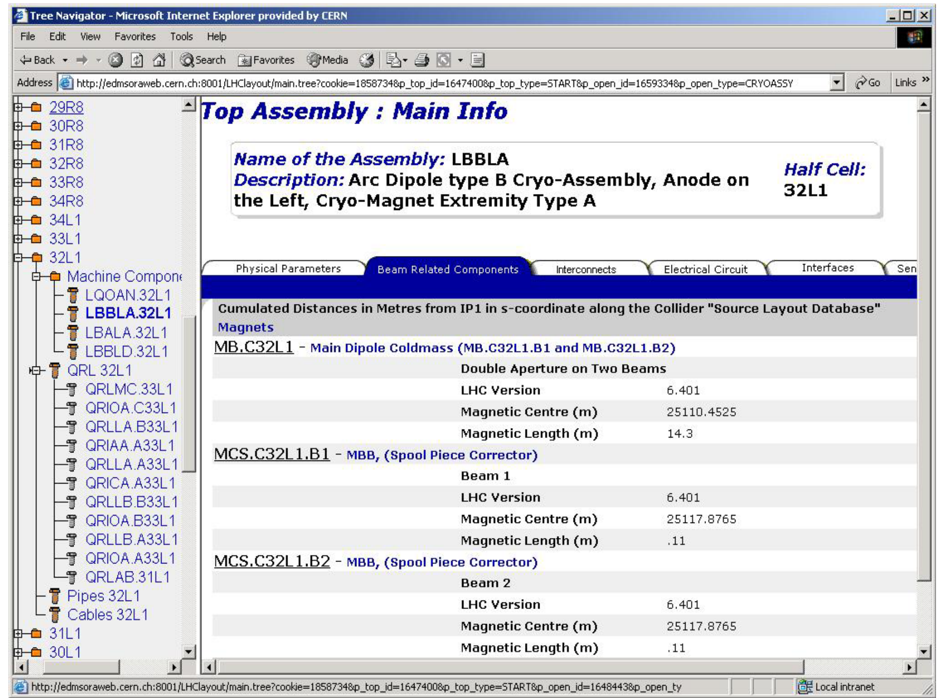
Ronny's point is…

- ☑ The *amount of information* in LHC is huge
- ☑ The relations between the information is multiple and depending on the user's point of view
- ☑ *Identification* is needed
- ☑ A naming/coding schema is a *convention*
- ☑ We can argue for hours… years and still not agree
- ☑ The "name" cannot be a *mnemonic* any more
- ☑ Existing names must be published and used as *reference*
- ☑ Related information for different usage must be *centralized*, *published* and *accessible*
- ☑ Names could be assembled from the info, *on the fly*…

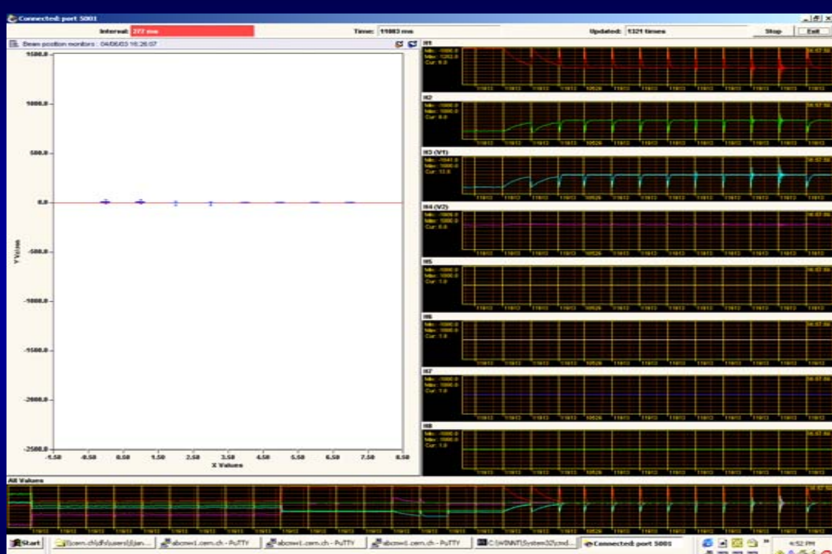
Existing Documents

Document	Title	Naming Example
LHC-PM-QA-206	LHC Part Identification	**HCMCSME001-AA000001**
LHC-PM-QA-204	Equipment Naming Conventions	**MCSBA.13L1.B1**
LHC-QI-QA-0002	Naming Conventions for Cryogenic Instrumentation	**MB.C32L1/TE821 (In cryo-assembly LBBLA.32L1)**
LHC-QRL-QA-0004	Naming Convention for the QRL	**QRLAC.E31R3/CV911**
LHC-DQ-ES-0003	Description of QPS Signals in LHC	**UExt, QD0,… from DQQDL**

This is the type of signal we want to get to in order to obtain information after a quench



Real-time



Overall Conclusions

- **A lot of hard work has gone in, responsibility has been accepted and a lot of progress has been made. These are critical systems:**
  - **Alarms**
    - **New architecture in place, integration challenge ahead…**
  - **Logging**
    - **Ready for some real data**
  - **Post Mortem**
    - **Requirements clear, manpower needed**
  - **Analogue acquisition**
    - **Looks Very Cool, NB 3-tier dependency**
  - **Timing**
    - **Grand unification, a lot of new hardware, challenges ahead**
  - **High level Controls: development, support, architecture and implementation.**
    - **Kicking and screaming into the 21st century**
  - **Middleware**
    - **A lot of progress on an agreed common solution**

Overall Conclusions

- Clear that the 3-tier architecture and the associated implementation on Oracle application server needs validation (as do other implementation choices).
- Use it appropriately i.e. choose the product with respect to the requirements
  - E.g 2000 lb gorillas like QPS
- Manpower issues abound… get defensive about what is taken on
- 24/7 Support will required…
  - Sonic MQ
  - Oracle application server
- Keep those milestones coming…

Thanks to all the speakers for a great set of very professional talks!

If I didn't know better, I'd almost think we knew what we are doing!!!