Pilot Project planning: ROB Complex
(R. Cranfield, UCL)

Aims

Current workplans and resources

What has to be done?

Functional areas to be covered

Technologies

Questions to be answered
Why is ROB studied by LVL2?

• Main raison d’être for ROB is LVL2 processing
• Some of the trickiest ROB constraints and requirements are associated with RoIR handling imposed by LVL2
• ROBs must interface correctly to the LVL2 network

BUT:

• ROB is also a key DAQ component
• ROB complex overlaps with DAQ front-end components

• Integration between LVL2 and DAQ is crucial!
Goals

• Produce documents (describing ROB options/choices)
• Produce hardware for next stage of prototyping?
• A year is not very long! Doesn’t allow many meetings.
• Build up documents via Web-based repository?
  – mailing list
  – main evolving document (maintained by editor)
  – comments/thoughts page
Points from Pilot Project overview

- Clear program and goals
- Map resources and commitments
- Integrate present and future work

- full LVL2 functionality
- technology candidates
- input to ROB URD
- RoI mapping
- data flow control
- RoI distribution

- Pre-processing
- special co-processing (TRT)
- networks
- software
Amsterdam workplan

1. a) Front-end data (incl pre-processing/formatting, XOFF, TTC)  Le Du
   b) Trigger strategies & menus  Blair, Strong, George, Bock, Hubbard, Clarke, Amadon, Bystricki
   c) Modelling  Vermeulen

2. DAQ/EF liaison  (esp. monitoring)  Cranfield

3. Work in progress  (protoypete development; multi ROB-in tests; S-link (Paroli, etc); DAQ -1 tests)

4. “Theoretical” studies (not discussed in Amsterdam)
Resource matching  
(provisional)

CERN  <  S-Link  
  optical links

SACLAY

MANNHEIM  <  ROIC (TRT processing)  
  FPGA generic

UK

NIKHEF  <  multi-ROB modelling  
  optical links  
  hardware v. software (SHARC design)

DAQ  <  DAQ -1 ROB component prototyping  
  multi-ROB-in
What is a ROB?

LVL2 functional view  
(after Strong)

MultiROB concept  
(after Le Du)
Components

- Link drivers/receivers
- Buffer memory
- Event fragment manager/selector
- LVL1/Supervisor interface (RoI receiver/distributor)
- Pre-processor
- LVL2 interface
- LVL3 interface
- Monitoring/error/status interface
- Run-control interface
- Inter-component interfaces
- Test facility
Technology

• **Technology prototyping:**
  – Optical links
  – FPGA
  – Fast processors
  – PCI
  – VME
  – Blink
  – DSP link
  – ATM
  – SCI
  – Gigabit Ethernet

• **Technology watch:**
  – Data input
  – Logic implementation
  – Memory
  – Memory management
  – Inter-module communication
  – Processing power
  – Network I/O
Questions I

ROB functionality

1.1 What event types must a ROB recognise and how should it handle them?
1.2 How big must a ROB buffer be to achieve acceptable deadtime?
1.3 What is the likely distribution of event fragment sizes in each sub-detector region?
1.4 Is it necessary to have an ID (e.g. LVL1 trigger number) for each event? If so, what is this ID when LVL1 triggers are not available?
1.5 What is the likely distribution of LVL2 latencies?
1.6 How should full buffers be handled?
1.7 Are there constraints on the issuing of XOFF on the ROLs?
1.8 Should a ROB communicate directly with the LVL1 trigger processor? If so, how?
1.9 Do the ROBs need TTC info? If so, how should it be handled?

1.10 How should the ROBs interact with run-control?
1.11 How should the ROBs interact with error-reporting?
1.12 How should the ROBs interact with status-reporting?
1.13 What test facilities are required?
Questions I (...contd)

ROB functionality

1.14 How should primary RoI requests be distributed?
1.15 How should secondary RoI requests be distributed?
1.16 How should LVL2 decision data record be stored?
Questions II

ROB grouping

2.1 Should all ROBs have identical hardware?
2.2 Should all ROBs have identical software?
2.3 How should the failure of a ROB component be handled?

2.4 Can the URD constraint on ROD latency be met and does this impact the ROB design?
2.5 Is the number of ROBs per sub-detector optimum (and scalable to 100kHz) and does this impact the ROB design?
2.6 Is there an optimum grouping of ROBs? If so, does the size vary greatly from sub-detector to sub-detector?
2.7 Is it satisfactory to do the TRT scan in parallel? If so, what is the minimal ROB grouping?
2.8 How might the predicted HCAL ROB bottleneck be removed? More ROBs? Ignore in first step? Head-of-line blocking avoidance techniques?
2.9 Is it desirable to pre-process data merged across sub-detectors? If so, can this be done in a ROB complex?
2.10 Does the possibility of an SCT scan affect the choices for ROB grouping?
Questions III
Technology options

3.1 How much board space will ROB components require?
3.2 What is optimum packing arrangement? (PMC, 6/9U VME, PCI, ASIC,…?)
3.3 What are the likely constraints imposed by crosstalk?
3.4 What are the likely constraints due to power requirements and dissipation?
3.5 What is the likely achievable hardware logic capacity?
3.6 What is the likely achievable processor speed?
3.7 What is the likely achievable memory capacity?
3.8 What is the likely achievable memory speed?
3.9 What is the likely achievable speed for inter-module communication?
3.10 What is the predicted commercial availability of components?
3.11 What is the estimated cost of components?
3.12 What is the predicted reliability of components?
3.13 How easy will it be to maintain components? (accessibility, swapability, etc)
Next steps

1) Agree documentation mechanism (Web, email)

2) Agree documentation framework (sub-headings)

3) Compile documentation checklists (issues, questions)

4) Compare current workplans with checklists

5) Agree revised workplans