ROB UML
for ATLAS Level 2 Trigger

Restricted to level 2 trigger activity
No Event Builder    (See DAQ Group : D.F.)
Methodology

Describe requirements  Use Cases

Identify the Actors
Describe their requirements from the system

System Analysis  for each Use Case

Identify the Objects
Description of their Classes
Relations between Classes
Interaction between Objects
Messages

Logical View

Class Diagram
Object Diagram
Sequence Diagram
State Diagram

Design

Same Diagrams
+ Component Diagram
Deployment Diagram

Main Actors

Main activity:  Data Flow

Actors:

ROD

delivers data to the ROB
receives a busy signal when ROB is almost overflowed

ROB Collector

receives from a feature extractor a request based on RoI
gets the corresponding ROBs
requests complete or part of event fragment to the ROB

Level 2 Supervisor

requests deletion of event fragment to the ROB
delivers event numbers to be kept
Main Actors

Peripheral activity: Run control

Nothing specific to the level 2
same interaction as for the DAQ initialisation of parameters
load of data for test

Monitoring

provides statistics on data and requests management
on ROB behaviour

No monitor on data

Remarks

ROBs are embedded in Nodes of a Network

Make a clear separation between
the actual ROB functionnality
and
the message management

reception of requests
dispatch of requests
returned data in messages

Buffer management

Page organisation and Fifos for page management are specific to
some implementations

cf: circular buffer

Preprocessing

Are there specific requests to preprocess data?

Probably not,
but data delivered can be more or less prepared for feature extractor
according to different detectors
set at configuration time??
Requirements for the ROB

at that level:

ROB = entity able to store and retrieve data on request

Do not care of the internal organisation

Use Diagram for the Data flow

ROB

Store Events

Request complete or part of event fragment

Delete Events

Accept Events

??

Feature Extractor / Rob collector

L2 Supervisor

ROD
Use Diagram for other activities.

Store events
### Request RoI / event fragment

**Rob Collector**
- Request RoI
  - get event fragment
  - extract RoI
- Returns RoI

**Rob**
- Request event fragment
  - get event fragment
- Returns event fragment

### Delete / Accept event fragment

**L2 Supervisor**
- list of event fragments to be deleted
  - delete event fragments

**Rob**
- list of accepted event fragments
  - ?
Run control

: Run Control

: Rob

- initialise
- get parameters
- load data for test
- get data for test
- start
- pause
- resume
- stop

acknowledgement from Rob?

Monitoring

: Monitor

: Rob

Any time in the run
- get statistics
- partial statistics

End of the run
- get statistics
- complete statistics
Store event fragment

Initialise store for new event

Store new event

Delete event fragment

Request event fragment

ROB Manager could also be a request dispatcher
Next Step

Define classes corresponding to previous objects
Study the relations between the classes

ROB Complex

Several ROB(in) can be grouped in a ROB Complex for:

embedding one (or several) ROB(s) in a Node of the L2 system
adapt the throughput of the output to the network capability (reduce cost)

ROB Complex is considered as a macroRob
same interface as previously defined
with a messaging mechanism to receive and transmit messages from network(s)
External connection of the ROB Complex

ROB Complex organisation

ROB UML for L2

L2-DAQ meeting CERN 4 June 1999

M. Huet Saclay
ROB To Switch Interface organisation

ROB Controller organisation
**ROBIN organisation**

*(example based on event buffer organised in pages)*

- **input data manager**
  - raw data
  - **derandomizer**
    - raw data
    - uses
    - **event manager**
      - raw data
      - uses
      - **event table**
        - raw data
        - describe
      - **event store**
        - organised in
        - **page**
          - uses
          - **free page**
          - **used page**

- **request manager**
  - mailbox
    - deposit requests
  - mailbox
    - deposit data
    - address of data

- **Controller**
  - **ROB**
    - mailbox
      - deposit requests
    - mailbox
      - deposit data
      - address of data

- **event manager**
  - event table
    - 1
    - event store
      - 1
      - mailbox
       - deposit requests

- **Rol extractor**
- **input data manager**