

# “Boundary” Specification for component: SIMPLE ROB

**Authors : P.Clarke, G.Crone, W.Li**

## *Abstract*

*This provides the high level specification of a simple ROB. This means in effect a single ROBIN+ROBOUT+ROBController in one unit.*

---

Version : 2.1

Date : 13-1-98

Reference :

---

## 1 Introduction/Notes

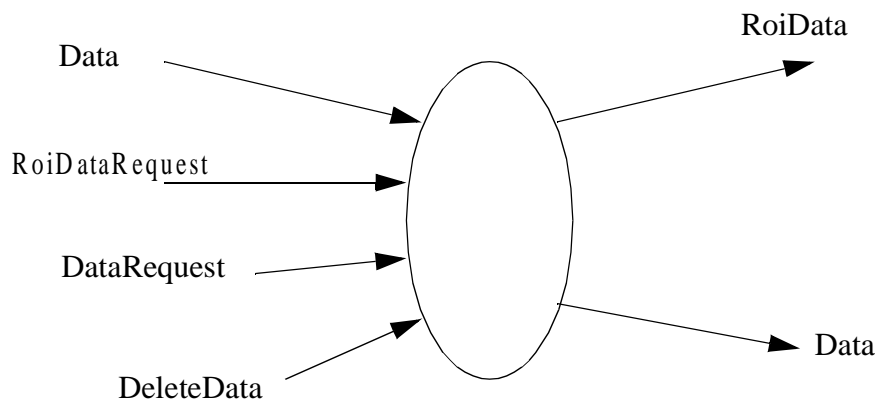
Note: It is to be understood that the following information is implicitly included in all messages:

**length**

**source & destination addresses**

and is therefore not shown explicitly in the message descriptions below.

## 2 Message summary diagram



## 3 Messages in

---

***Message name : Data***

This message signifies that a block of data has been transferred into the ROB. It is sent by a data supplier (probably the event generation module, but could also be a ROD).

***Contents {***

```
    eventIdentifier
}
```

**Responsibilities:**

1. Upon reception of this message the ROB should note that it contains this data block.
2. Store this message as it will be requested as an output.
3. If an **RoiDataRequest** message for this event is already present in the node then cause the **RoiData** message to be sent at a time as determined by the specific ROB implementation. Following this the **RoiDataRequest** message should be deleted.

---

**Message name : RoiDataRequest**

Message sent to ROB to request event information corresponding to a specific ROI

**Contents {**

eventIdentifier

roiIdentifier

*A unique identifier within an event for this roi. (Could be contained within roiDescriptor but for now left as separate item)*

roiDescriptor

*A descriptor (exact form yet to be determined) which contains any roi specific information needed to carry out duties.*

targetAddress

*Address of target node to which Roi data is to be sent.*

**}**

**Responsibilities:**

1. Upon reception of this message the ROB should note that it contains this unsatisfied request.
2. If the **Data** message for this event is already present in the node cause the **RoiData** message to be sent to the targetAddress at an appropriate time. Following this the **RoiDataRequest** message should be deleted.

---

**Message name : DeleteData**

Message sent to a ROB to inform it to delete a list of events.

**Contents** {  
    listof<eventIdentifier>  
        *List of identifiers of events to be deleted*  
}

**Responsibilities:**

1. Delete each event in the list from this node. Following this the **DeleteData** message should be deleted.

---

**Message name : DataRequest**

Message sent to a ROB to ask for entire event data to be sent out. It is an error condition if this ever arrives before the data is present.

**Contents** {  
    eventIdentifier  
  
    targetAddress  
        *Address of target node to which event data is to be sent.*  
}

**Responsibilities:**

1. Cause the **Data** message to be sent to the targetAddress at an appropriate time. Following this the **DataRequest** message should be deleted.

## **4 Messages out**

---

**Message name : RoiData**

Message sent by ROB to simulate Roi data being sent out following an **RoiDataRequest**.

**Contents** {  
    eventIdentifier  
    roiIdentifier  
}

---

**Message name : Data**

Message sent by ROB following **DataRequest** message. This requires the information in the stored data message to which it refers to be forwarded without alteration (apart from addressing).

**Contents** {  
    *<defined to be identical to the original **Data** message received by ROB>*  
}