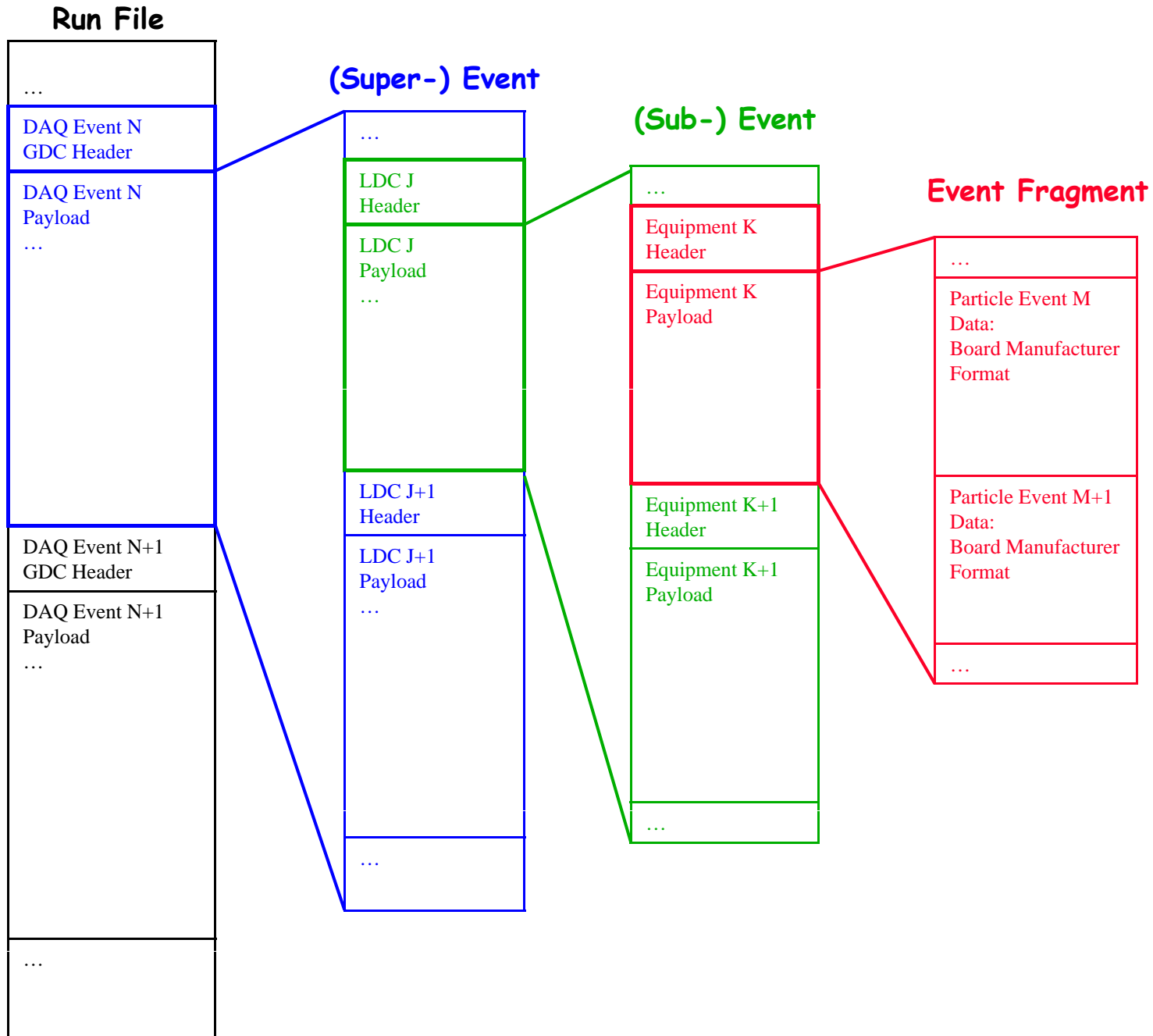


---

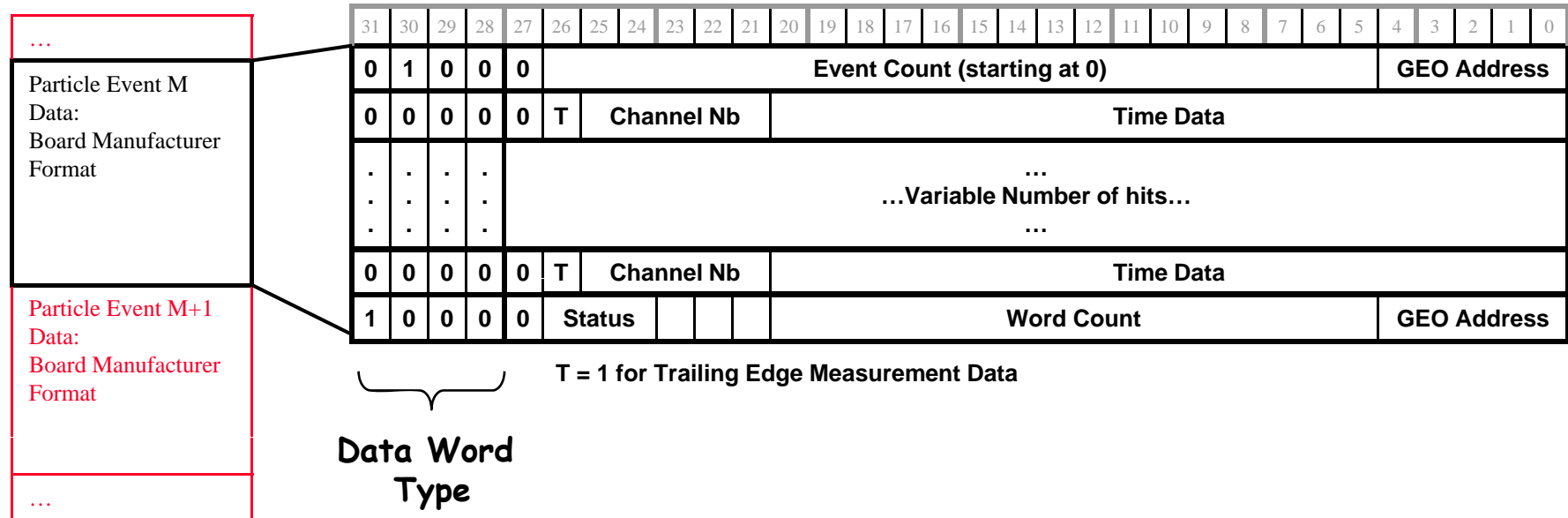
# Unpacking Data for Online Monitoring

Jean-Sebastien Graulich, Geneva

- o DATE Data Format
- o DATE monitoring package
- o MICE Online Monitoring Project
- o Unpacking Classes
- o Plans



# Example of Particle Data Format: TDC V1290



**Event Header**

Event Size
Sync. Word
Header Size
Header Version
EventType
RunNb
Event Id[0]
Event Id[1]
TriggerPattern[0]
TriggerPattern[1]
DetectorPattern[0]
DetectorPattern[1]
Attribute[0]
Attribute[1]
Attribute[2]
LDC Id
GDC Id
TimeStamp[0]
TimeStamp[1]

**Same header for Super- and Sub-Events (only the attribute value is different)**

**Equipment Header**

Equipment Data Size
Equipment Type
Equipment User Id
Equipment Attribute[0]
Equipment Attribute[1]
Equipment Attribute[2]
Equipment Word Size

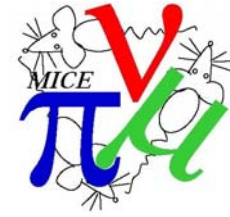


**Conventional Table of Equipment Type:**

Type	Equipment
0	Random Generator
100	V2718
101	Trigger Receiver
102	TDC V1290
103	fADC V1724
104	VLSB
110	Trailer
111	Scaler V830



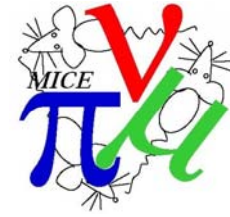
# DATE Monitoring package



- ◆ **Date Provide a monitoring library allowing**
  - Accessing Events on TCP/IP Sockets or from File
  - Filtering Events (based on Event Type)  
E.g. Physics Events or Calibration Events
  - Defining Policies  
E.g. Get all events or only a subset
- ◆ **Main user API function**
  - `monitorGetEvent( void * Buffer)`  
Allocates memory to Buffer and set it pointing to the first word of the next event
- ◆ **Main user application**
  - `eventDump datafile.dat`  
Print the content of the binary data file in ascii format



# MICE Online Monitoring



## ◆ Aim

- Produce Online Monitoring Histograms/Root Trees using the DATE monitoring Package

## ◆ Main Steps

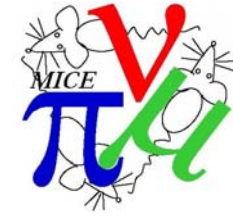
- Unpacking
- Filling Histograms
- GUI

## ◆ Main Constraint

- Unpacking software should be reusable in G4MICE for Online Reconstruction  
=> We want to do it in C++



# Unpacking Classes



```
class MDdataContainer {
protected:
  unsigned char* _data;
  unsigned int   _size;
  bool           _valid;

public:
  MDdataContainer( void *d=0,unsigned int s=1 ):_data( (unsigned char*) d ),_size(s),_valid( false ) {}

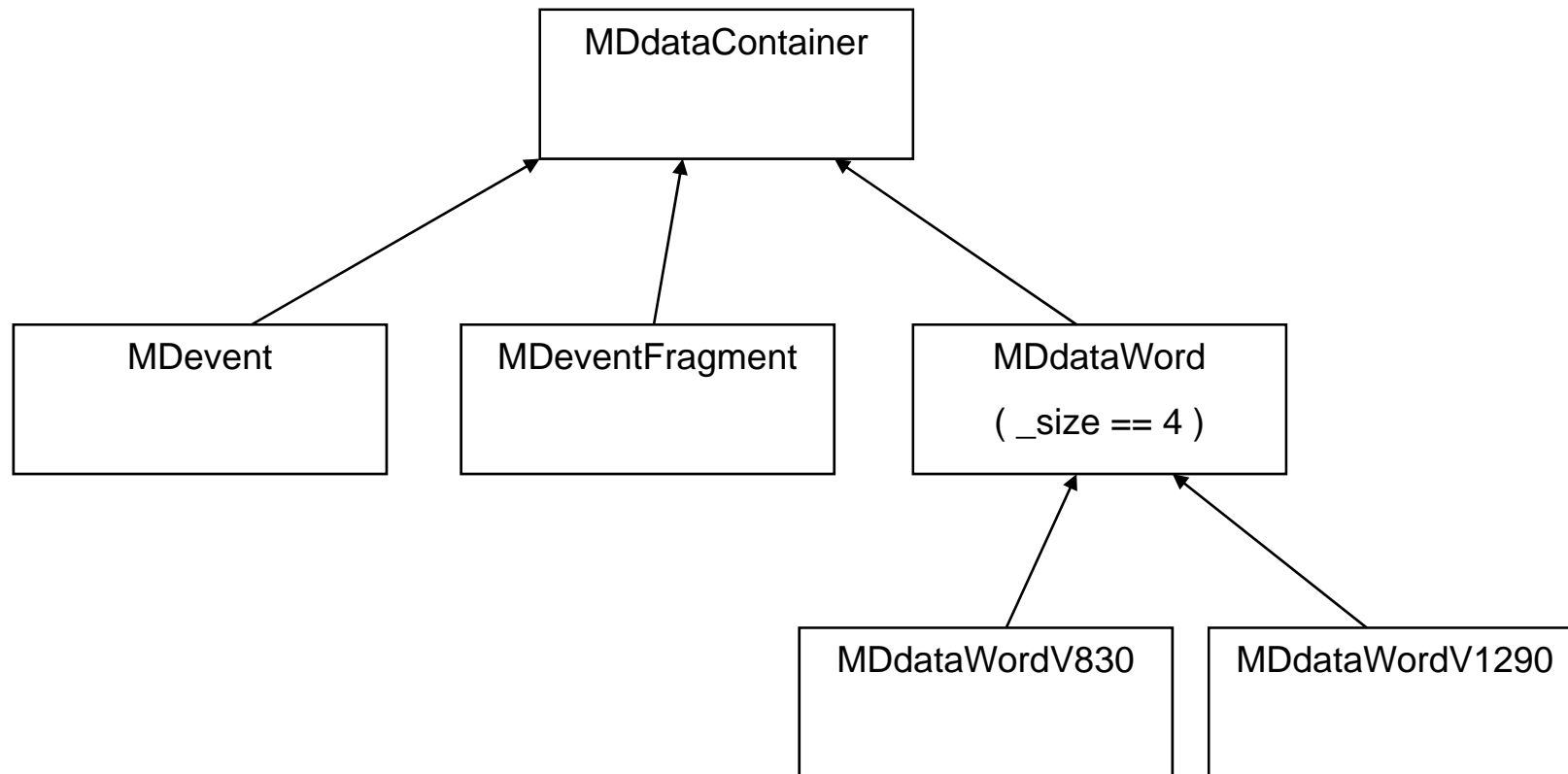
  MDdataContainer( MDdataContainer& dc ) ;
  virtual ~MDdataContainer(){}

  unsigned char* GetDataPtr( unsigned int i=0 ) ; unsigned long32* Get32bWordPtr( unsigned int i=0 ) ;
  bool           IsValid( void ){ return _valid; }
  unsigned int   GetSize() { return _size; }

  virtual void SetDataPtr( void *d );
  void Validate()   { _valid = true; }
  void UnValidate() { _valid = false; }
  void SetSize(unsigned int s) { _size = s; }

  virtual void Dump( int atTheTime=1 );
};
```



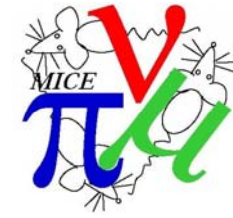


**Each class**

- Has its own unpacking member functions
- Has a `Dump()` virtual member function



# Plans



- ◆ New classes *MDparticleEventEquip*
- ◆ Implement member functions to fill histograms
  - → dependency to ROOT !
    - + use Compilation flag ?
    - + Change design ?
    - + give up on sharing the unpacking with G4MICE ?
- ◆ In //: Implement other equipments: fADC and VLSB