



Go4



GSI
Online
Offline
Object
Oriented

The GO4 Analysis Framework

J.Adamczewski, M.Al-Turany, D.Bertini, H.G.Essel

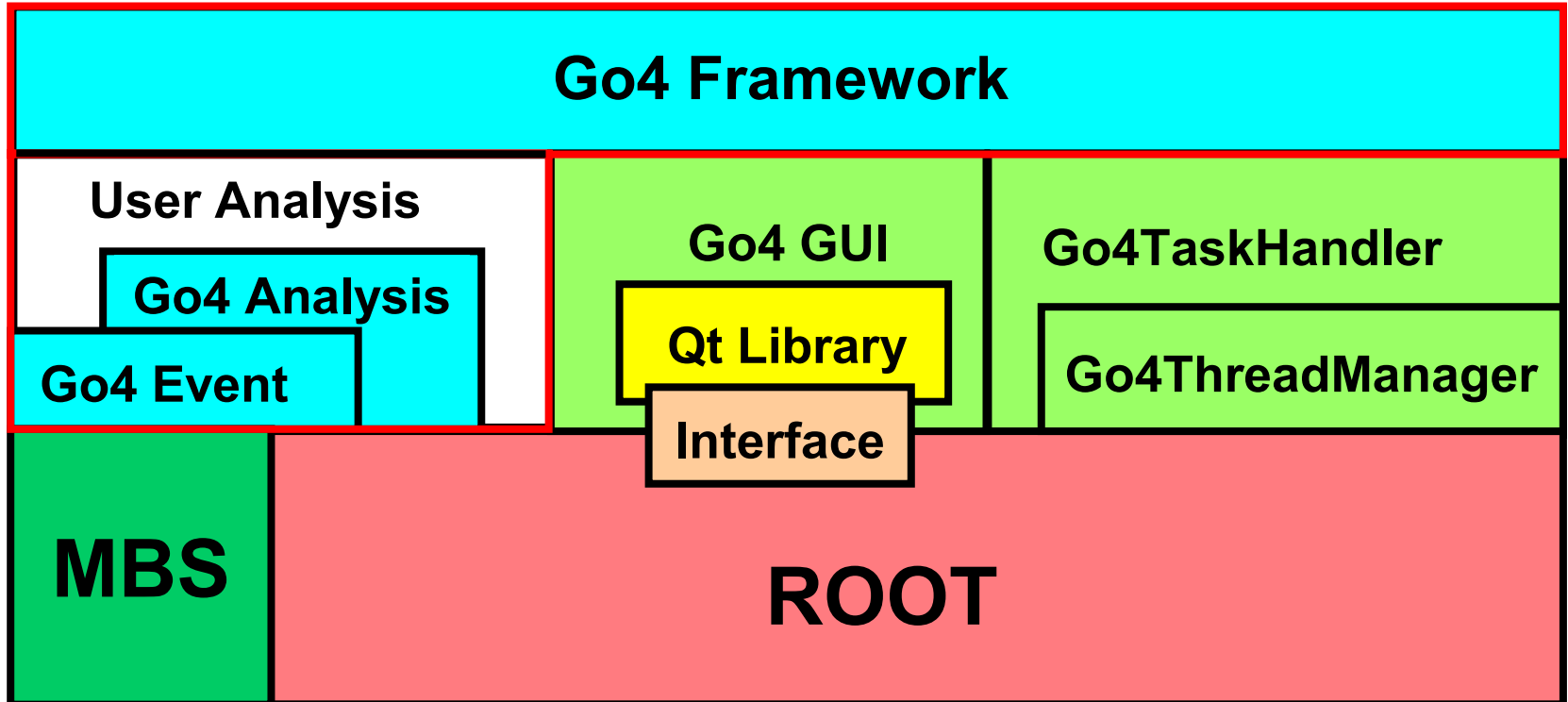


Contents

- **Go4: Software layers** and architecture
- **Go4 Analysis framework concepts**
- **Base classes** for user event structures
- **Example:** Euroball
- **Exercises:** change analysis run configuration
- **Go4** in batch mode



Go4 Package Layers





Go4 Framework concepts

- Based on **standard ROOT** system
- **Online** and **Offline** Framework
- **Batch mode** and **interactive mode** (GUI)
may run same analysis
- **Any user analysis** may be adopted

Go4 provides **interfaces** (base classes) to
„plug-in“ user event classes and analysis

Go4 implements all **general services**



Event Base classes

- **TGo4EventElement:**
Event structure (input and output event)
- **TGo4EventSource:**
Fills event structure with data
- **TGo4EventStore:**
Stores event structure
- **TGo4EventProcessor:**
Converts input event into output event;
subclass of TGo4EventSource
- **TGo4EventFactory:**
Defines the user implementations of all
the above at initialization



Event service classes

- Interface to **MBS**:
 - TGo4MbsEvent, TGo4MbsSubEvent (format 10,1)
 - TGo4MbsFile (read from *.lmd)
 - TGo4MbsEventServer
 - TGo4MbsStream
 - TGo4MbsTransport } (connect to MBS)
- TGo4RevServ (connect to remote event server)
- **Root File** implementation:
 - TGo4FileSource, TGo4FileStore (TTree in TFile)
 - TGo4BackStore (TTree in memory only)



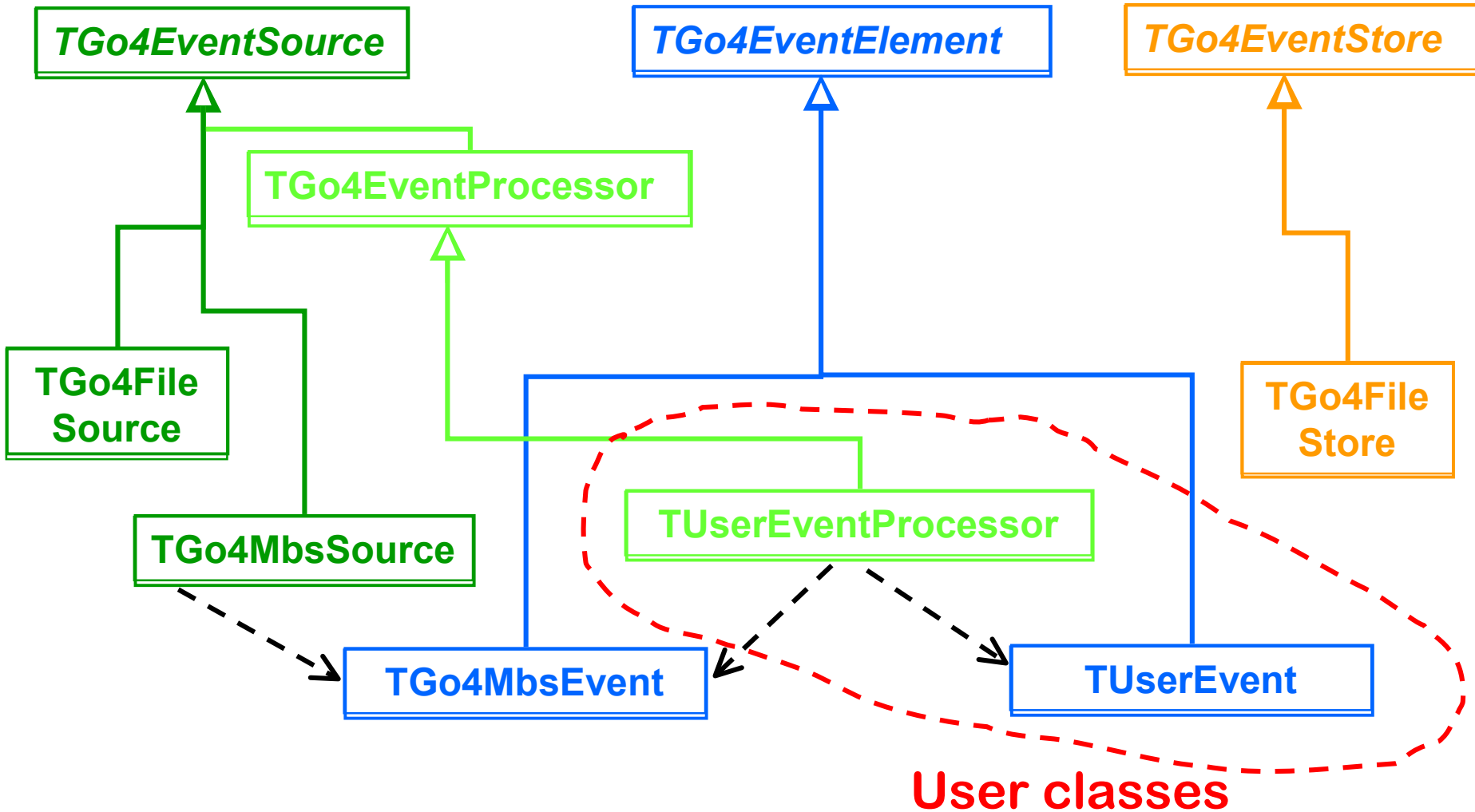
Event service classes

- **For more complex event structures:**
 - TGo4CompositeEvent
 - TGo4ClonesElement

< Denis Bertini on Wednesday



Event classes diagram





Example: reading from MBS

```
Short_t procidfield[2]= {1,2};
Char_t subcratefield[2]={0,5};
Char_t ctrlfield[2]={0,0};
UInt_t sizefield[2]={4,10};
TGo4MbsEvent* event = new TGo4MbsEvent(2, subcratefield, ctrlfield, procidfield, sizefield);
TGo4EventSource* input=0;
TGo4EventStore* output=0;
////////// MBS LISTMODE FILE
input= new TGo4MbsFile("/s/adamczew/ebtest.lmd");
////////// MBS TRANSPORT
// input= new TGo4MbsTransport("r2f-2");
////////// MBS STREAM
// input= new TGo4MbsStream("r2f-2");
////////// MBS EVENTSERVER
// input= new TGo4MbsEventServer("r2f-2");
////////// REMOTE EVENT SERVER
// input= new TGo4RevServ("r2f-2");
output = new TGo4FileStore("MbsEvents",1,5);
event->SetEventSource(input);
for(Int_t t=0; t<maxevents; ++t)
{
    event->Clear();
    Int_t ermess=event->Fill();
    //.. do something with event here
    // ...
    output->Store(event); // write to root file
}
```



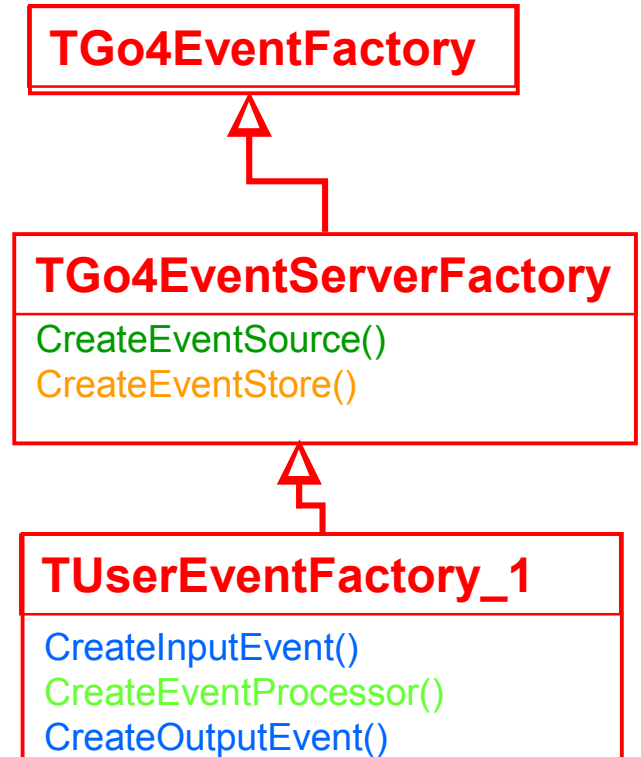
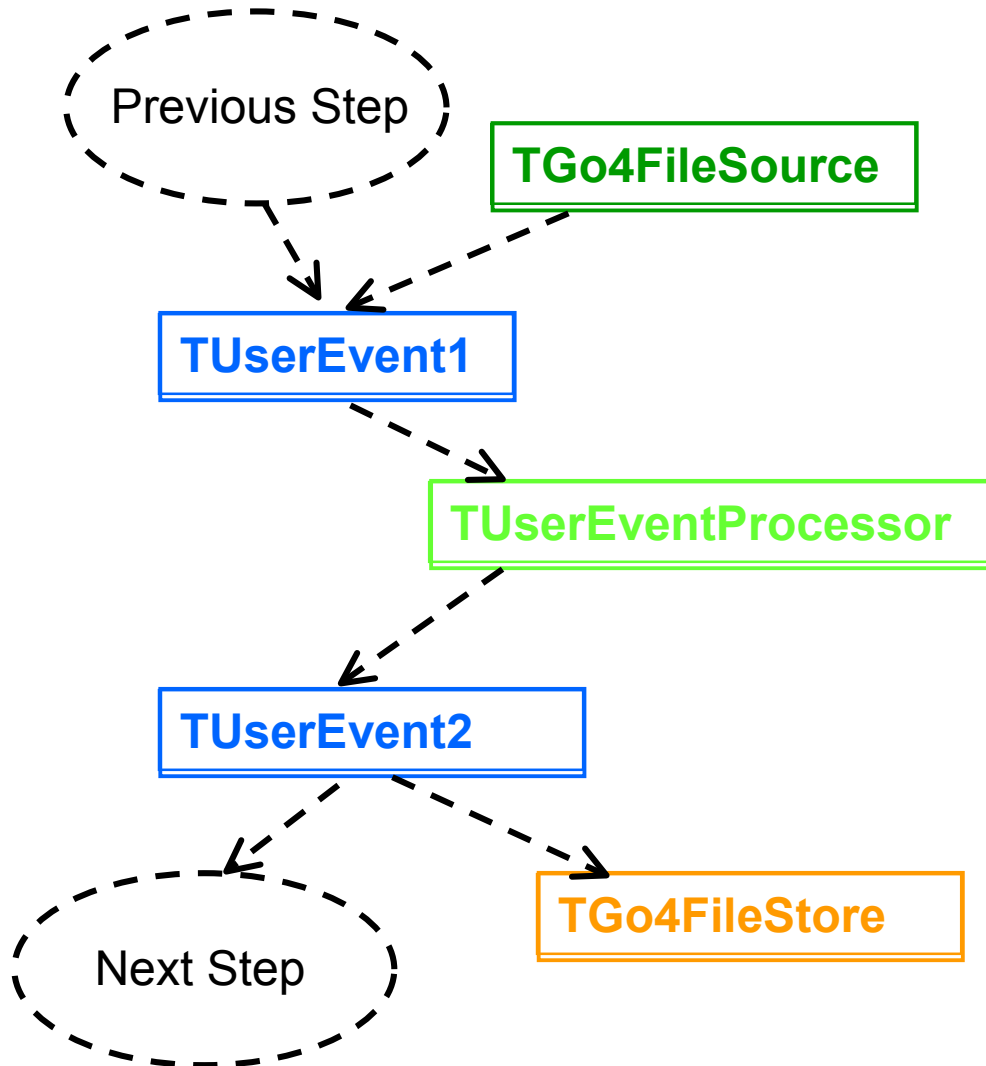
Analysis framework

- **TGo4Analysis** baseclass:
 - Analysis setup (chain of analysis steps)
 - Object organization, dynamic histograms
 - Implicit / explicit event loop
 - Virtual methods to be defined
in **user analysis subclass**
- **TGo4AnalysisStep**:
 - defines one stage of the analysis,
implements event classes

User designs own subclass of TGo4Analysis



Analysis step





Event Factory

```
#include "TEbCalibrateFactory.h"
TEbCalibrateFactory::TEbCalibrateFactory(Text_t * name) : TGo4EventServerFactory(name){}
TEbCalibrateFactory::~TEbCalibrateFactory() {}
TGo4EventProcessor* TEbCalibrateFactory::CreateEventProcessor(TGo4EventProcessorParameter* par)
{
    TEbCalibrateProcessor* calproc    = new TEbCalibrateProcessor("EBCaliProcessor");
    TEbCalibration* calibration = new TEbCalibration();
    calproc->SetCalibration(calibration);
    return calproc;
}
TGo4EventElement * TEbCalibrateFactory::CreateOutputEvent()
{
    TEbCalibratedEvent* calievent = new TEbCalibratedEvent();
    SetOutputEvent(calievent);
    return calievent;
}
TGo4EventElement * TEbCalibrateFactory::CreateInputEvent()
{
    TEbEvent* event = new TEbEvent();
    return event;
}
```

TEbCalibrateFactory.cxx

Baseclass TGo4EventServerFactory delivers methods
CreateEventSource, CreateEventStore



Example: Eventstructure class

TEbEvent.h

```
TEbEvent : public TGo4EventElement
{
public:
TEbEvent();
virtual ~TEbEvent();
virtual Int_t Fill(); ← calls TEbEventProcessor::BuildEbEvent(this)
virtual void Clear(Option_t *t="");
static Int_t fgiZeroField[41];
Int_t fiT0; ←
Int_t fiT1;
Int_t fiT2;
Int_t fiT3;
Int_t fiT4;
// ...
// ... further channels with experiment typical names ....
// ...
Int_t fiA21;
Int_t fiA22;
Int_t fiA23;
Int_t fiA24; ←

ClassDef(TEbEvent,1)
};
```

data structure field



Example: User Eventprocessor

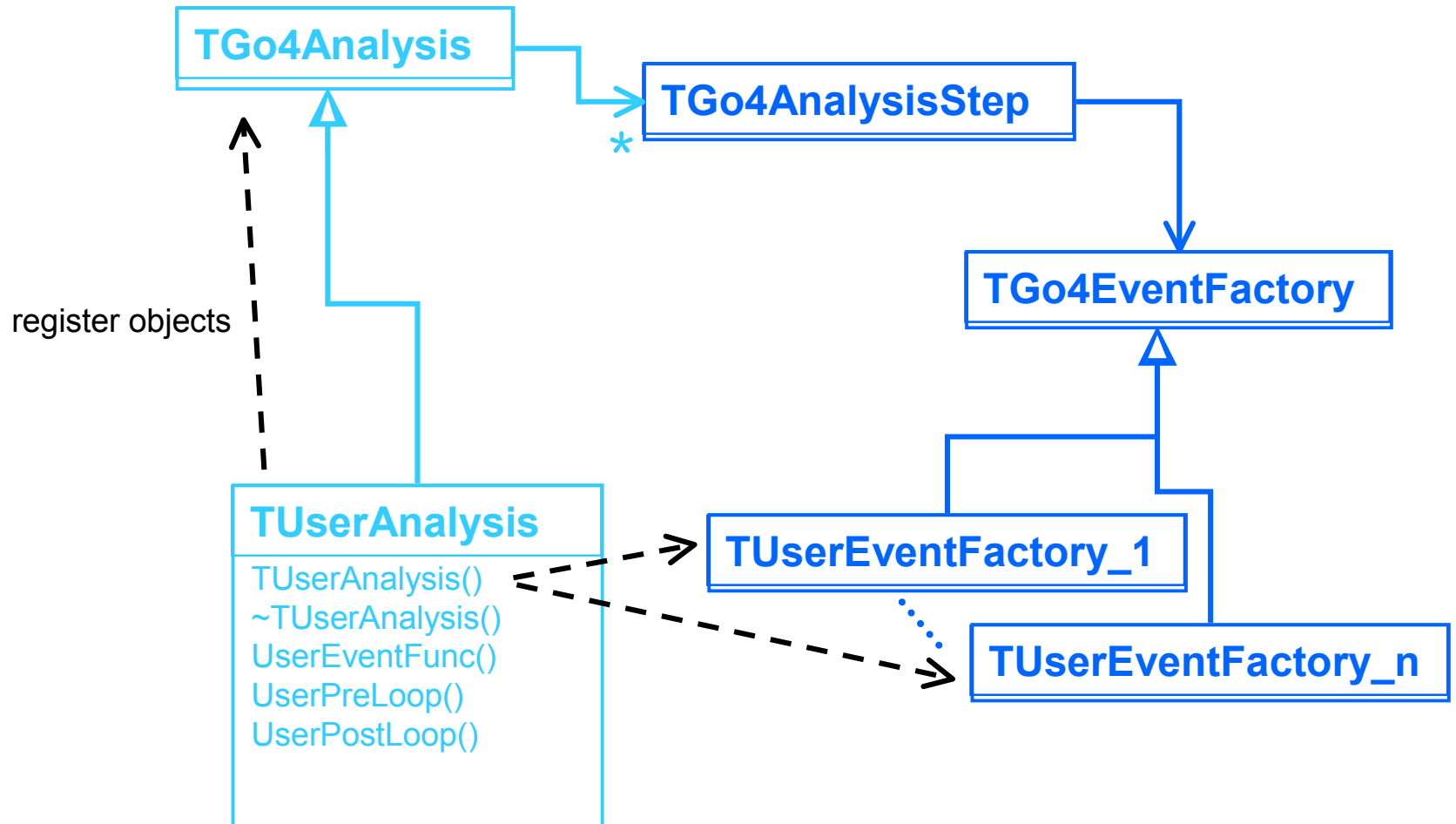
TEbEventProcessor.cxx

```
void TEbEventProcessor::BuildEbEvent(TEbEvent* target)
{
  TGo4MbsEvent*      input = dynamic_cast<TGo4MbsEvent*> (GetInputEvent());
  TGo4MbsSubEvent*  insub=0; Int_t index=0; Int_t value=0;
  if(input) {
    input->ResetIterator();
    while ( (insub= input->NextSubEvent() ) !=0 ) {
      if(insub->GetProcid()==1 && insub->GetControl()==9 && insub->GetSubcrate()==0) {
        target->Clear();
        Int_t* datafield    = insub->GetDataField();
        Int_t* outputbase  = &(target->fiT0);
        Int_t fieldsize    = (insub->GetDlen() -2) * sizeof(Short_t)/sizeof(Int_t);
        for(Int_t i = 0; i<fieldsize; ++i)
          {
            index      = *datafield&0xff;    /* from 1 to 40 */
            value      = (*datafield>>16)&0xff; /* from 0 to 4095 */
            if(index>40)
              cout << "index out of range, ignore!!!"<< endl;
            else
              *(outputbase+index)=value;
            datafield++;
          }
      }
      else if (insub->GetProcid()==1 && insub->GetControl()==7 && insub->GetSubcrate()==1) {
        // may do other things for other crate / detector part
      }
    }
  }
}
```

} unpack procedure



Analysis framework





Example: User Analysis class

```
TEbAnalysis::TEbAnalysis() : TGo4Analysis()
{
// first step definitions:
TEbEventFactory*          factory1    = new TEbEventFactory("Step1factory");
TGo4MbsFileParameter*    sourcepar1  = new TGo4MbsFileParameter("/s/adamczew/ebtest.lmd");
TGo4FileStoreParameter*  storepar1   = new TGo4FileStoreParameter(„eb-raw.root");
TGo4EventProcessorParameter*  procpa1  = 0 ;
TGo4AnalysisStep* step1 = new TGo4AnalysisStep("Unpack", factory1, sourcepar1, storepar1,procpa1);
AddAnalysisStep(step1);
// further analysis steps....

// register user objects to framework:
fxConTest1 = new TH1D ("ConditionTest1", "With window condition",1000,0,200);
fxConTest2 = new TH2D ("ConditionTest2", "With window condition",50,0,200,50,0,200);
AddHistogram(fxConTest1);
AddHistogram(fxConTest2);

fxWinCond2= new TGo4WinCond("windowcondition2");
fxWinCond2->SetValues(50,70,90,120);
fxWinCond2->Enable();
fxWinCond2->Invert(kTRUE);
AddAnalysisCondition(fxWinCond2);

}

```

TEbAnalysis.cxx



Example: User Analysis class

```
Int_t TEbAnalysis::UserPreLoop()
{
    fxMbsEvent=dynamic_cast<TGo4MbsEvent*> ( GetInputEvent("Unpack"))
    fxCaliEvent=dynamic_cast<TEbCalibratedEvent*> ( GetOutputEvent("Calibrate") );
    return 0;
}

Int_t TEbAnalysis::UserEventFunc()
{
    Int_t value=0;
    if(fxMbsEvent != 0)    value = fxMbsEvent->GetDlen();
    if(value != 0)    fxMbsHisto->Fill(value);

    if(fxWinCond1 != 0 && fxWinCond1->Test(fxCaliEvent->fdQ2MeV))
    {
        if(fxConTest1 != 0) fxConTest1->Fill(fxCaliEvent->fdQ2MeV);
    }

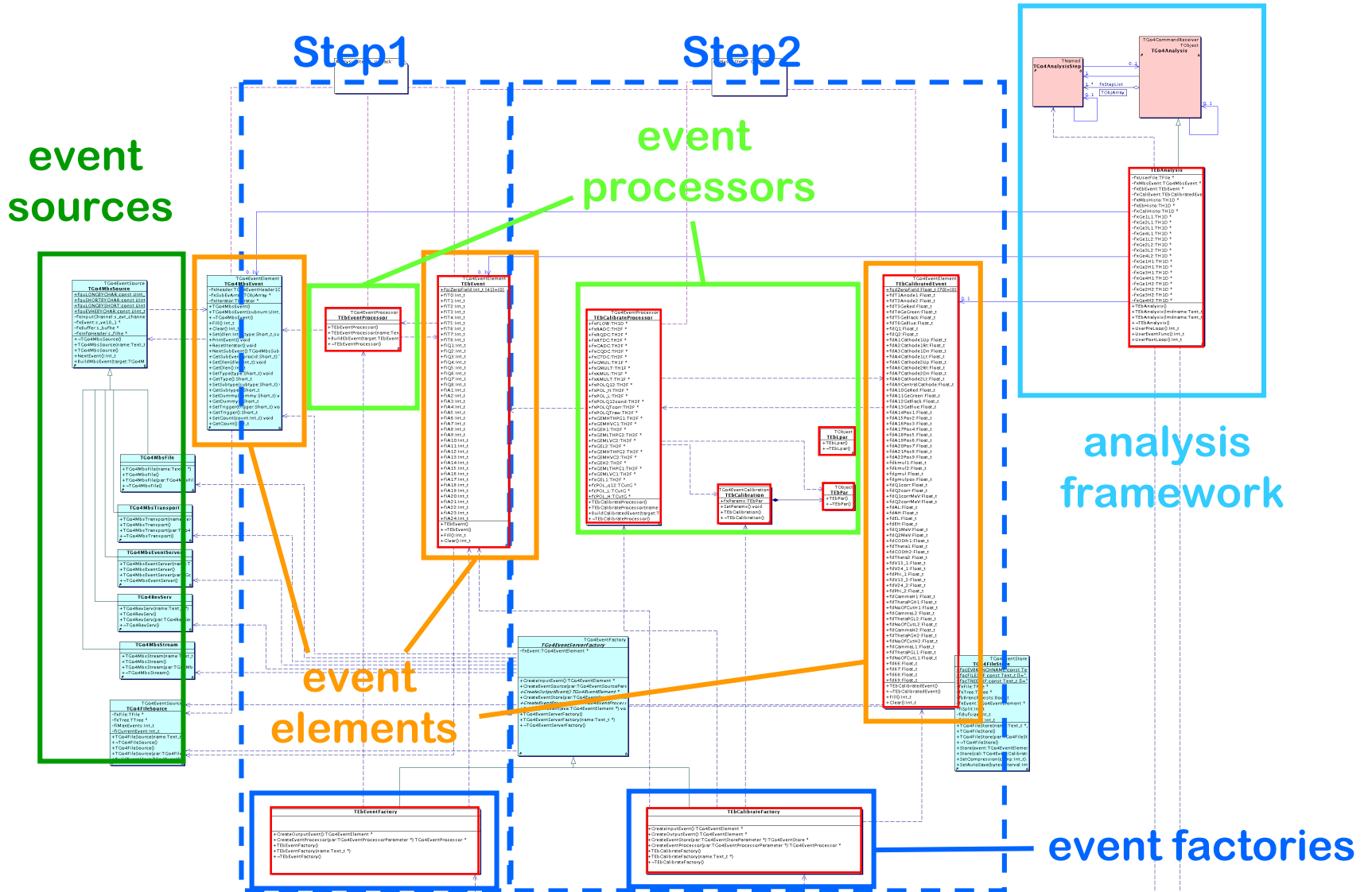
    //.....
}

Int_t TEbAnalysis::UserPostLoop()
{
    if(fxMbsEvent != 0)    fxMbsEvent->PrintEvent();
    fxMbsEvent=0;
}
```

TEbAnalysis.cxx



Example analysis: Euroball





Analysis configuration in GUI

- „. go4login“
- „cd go4/ws“
- „go4gui“ starts Go4 GUI
- Launch client
- Submit Settings
- Start Analysis



Analysis configuration in GUI

Exercise 1: change the EventStores of Steps:

- Set Output Filename of Step „Unpack“ to *FirstStepResult.root*
- Set Output Type of Step „Calibrate“ to TGo4BackStore (name can be left the same)
- Submit new settings
- Start Analysis
- Stop Analysis again after n events



Analysis configuration in GUI

Exercise 2: Disable second analysis step

- Disable Second analysis step „Calibrate“
- Set Output type of „Unpack“ to TGo4FileStore
- Set Output Filename of Step „Unpack“ to *FirstStepResult.root*
- Submit new settings
- Start Analysis
- Stop Analysis again after n events



Analysis configuration in GUI

Exercise 3: Disable first analysis step

- Enable Second analysis step „Calibrate“
- Disable First analysis step „Unpack“
- Enable Input of Step „Calibrate“
- Set Input type of „Calibrate“ to TGo4FileSource
- Set Input Filename of Step „Calibrate“ to *FirstStepResult.root*
- Set Output Type of Step „Calibrate“ to TGo4FileStore
- Submit new settings
- Start Analysis



User Analysis class

Exercise 4: Remove analysis step 2 from analysis

- Disconnect analysis from GUI
- Edit *TEbAnalysis.cxx*
- In constructor, comment line 121:
`//AddAnalysisStep(step2);`
- Make all
- clear old settings: „rm TGo4AnalysisPrefs.root“
- Connect new analysis to GUI
- Step „Calibrate“ should not appear anymore
- Submit new settings
- Start Analysis



Go4 in batch mode

- If running, Quit the Go4 GUI
- **„MainUserAnalysis /s/go4/ebtest 100000“**
- Start root CINT session: „root“
- Execute macro „.x browse.C“
- Use root TBrowser to view output files
- ROOT macros may be used for further analysis
- Output files may be examined later using the
Go4GUI DiskBrowser



Go4 in ROOTCINT

- Edit macro *ExplicitLoop.C*
- change line 28 to
 `gSystem->Load(„TGo4UserAnalysis.so“)`
- Start root CINT session: „root“
- Execute macro **„.x ExplicitLoop.C“**
- Precompiled Analysis with additional histograms
- Edit macro to change histogramming in loop



Using Go4 components

