



**Progress Report: CMS-UniAndes Group Meeting**

***An Emulator for the RPC Technical Trigger***

Andres Osorio-Oliveros

Universidad de los Andes

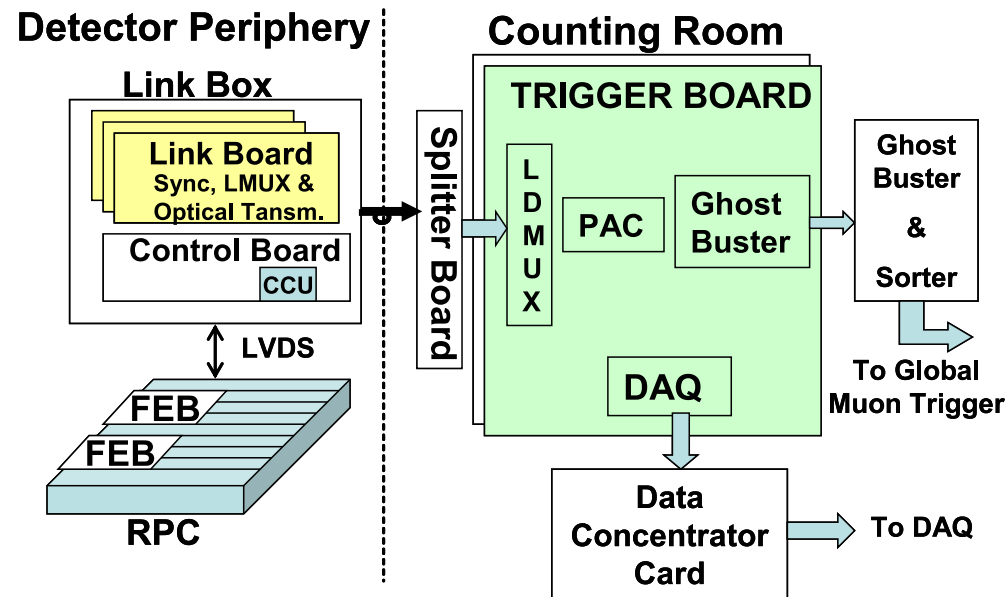
- Motivation
- RBC Technical Trigger
- Progress
- Next Steps

Sub-detectors at CMS have the ability to send special trigger signals known as **Technical Triggers**. These signals are used for test and calibration.

- The Resistive Plate Chambers or RPC are part of the Muon Trigger system.
- The RPC have implemented a Technical Trigger: the **RPC Balcony Collector (RBC) Technical Trigger**

The RBC-TT was designed in particular to deal with **cosmic muons** .

This is RPC Trigger Electronics full chain:

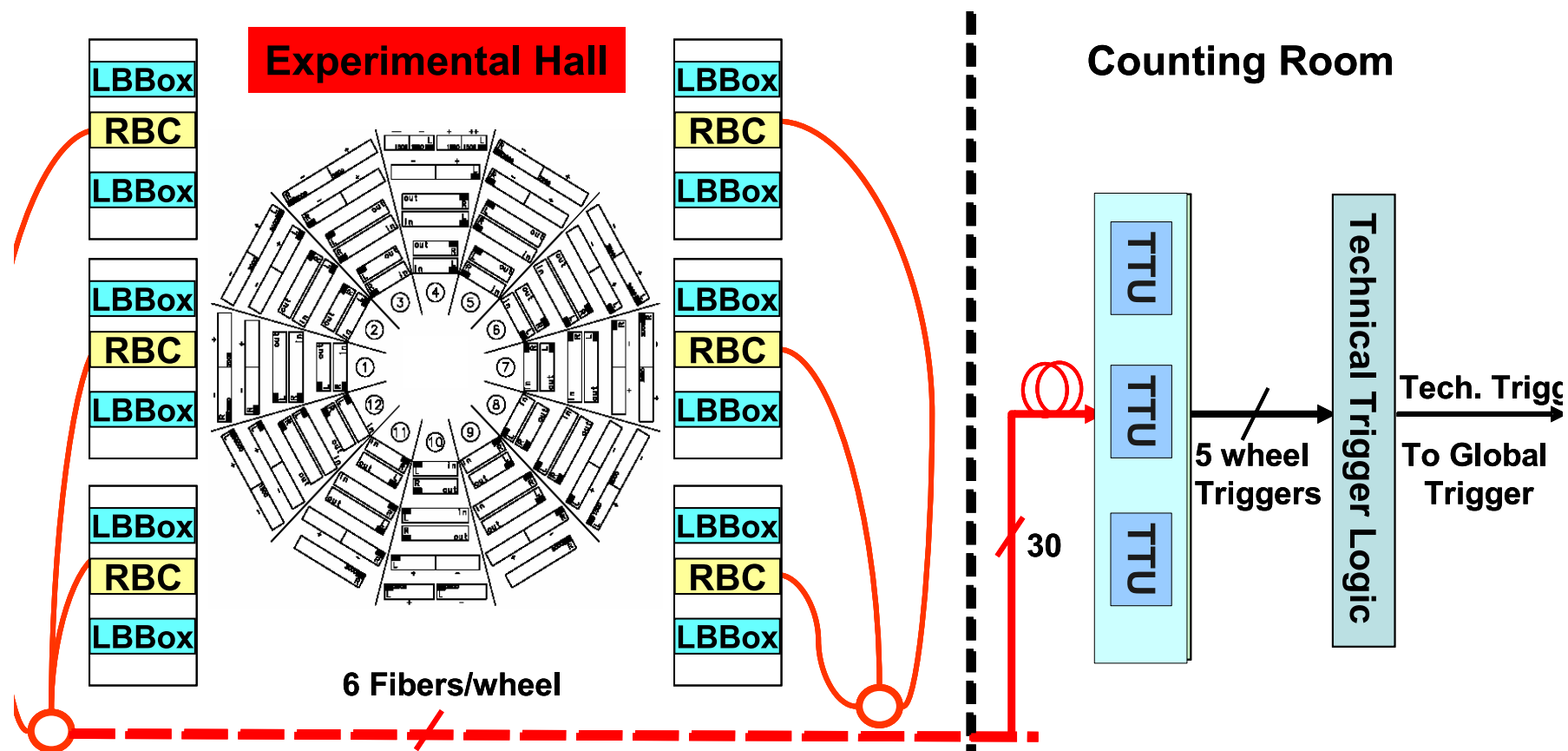


From F. Loddo (Ref [1])

- This system is optimised to trigger muons from the interaction point

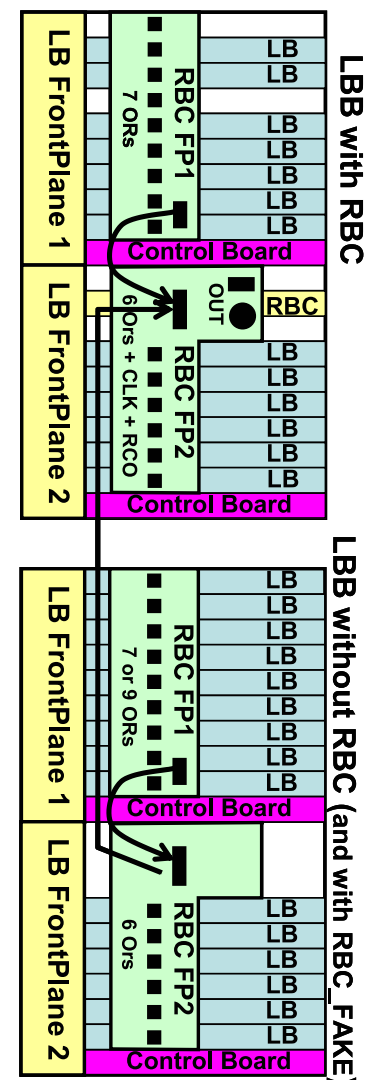
# RBC Technical Trigger

New boards were implemented:



From F. Loddo (Ref [1])

- RBC devices housed in the LB boxes: a total of 30 boards are used to cover the whole barrel (2 sectors per board)
- TTU ( $\equiv$  RPC Trigger Board with modified firmware): installed in the RPC Sorter Crate
- 1 Technical Trigger Unit or TTU each one has 2 Pattern Comparator (PAC) electronics: 1 PAC is used per Wheel

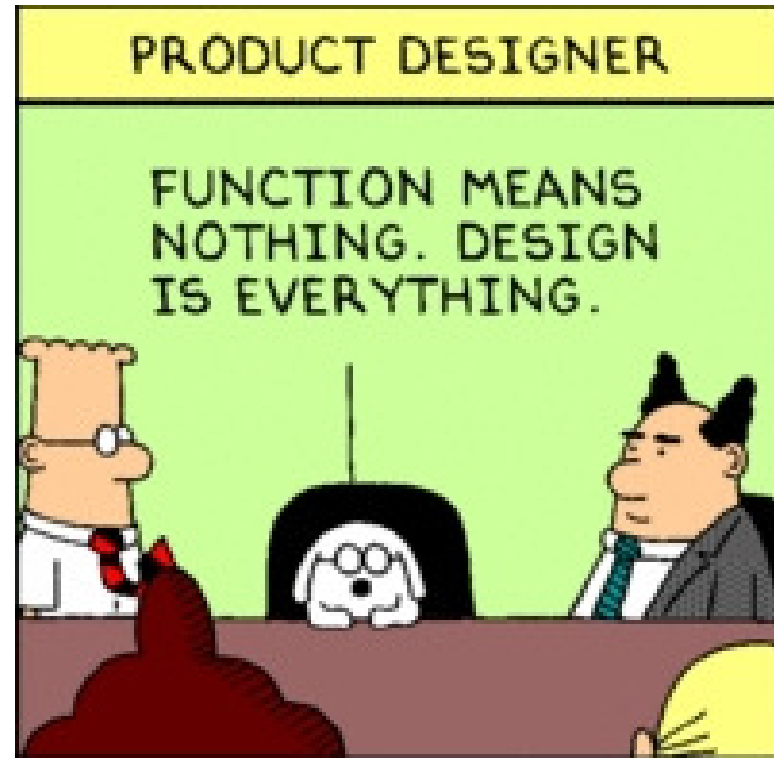


From F. Loddo(Ref [1])

The project consists on the full implementation of an emulator for the RBC Technical Trigger (**RBC logic + Technical Trigger Units Logic**).

Definition of an Emulator:

- An emulator duplicates the functions of one system using a different system, so that the second system behaves like (and appears to be) the first system. This focus on exact reproduction of external behavior.(from Wikipedia)
- → software interpretation of an electronic computer
- emulation  $\neq$  simulation
- **In CMS: Useful because in this way we can test and debug the system being emulated**
- Basically, we should be able to reproduce the results from a given pattern



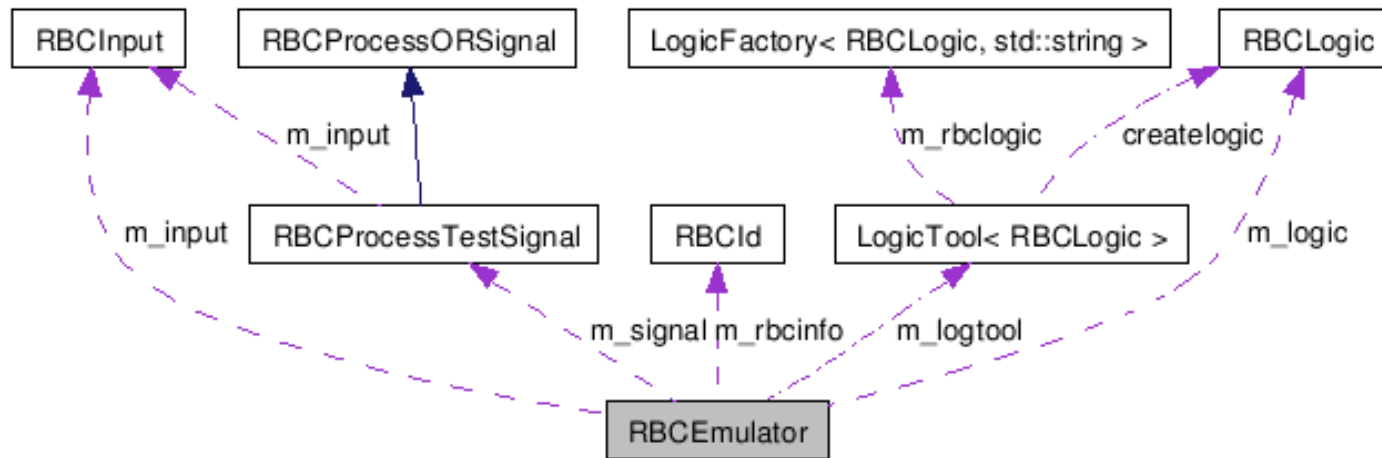
- The emulator has to be generic and flexible enough to include deferent logic implementations



# Implementation: Class Collaboration



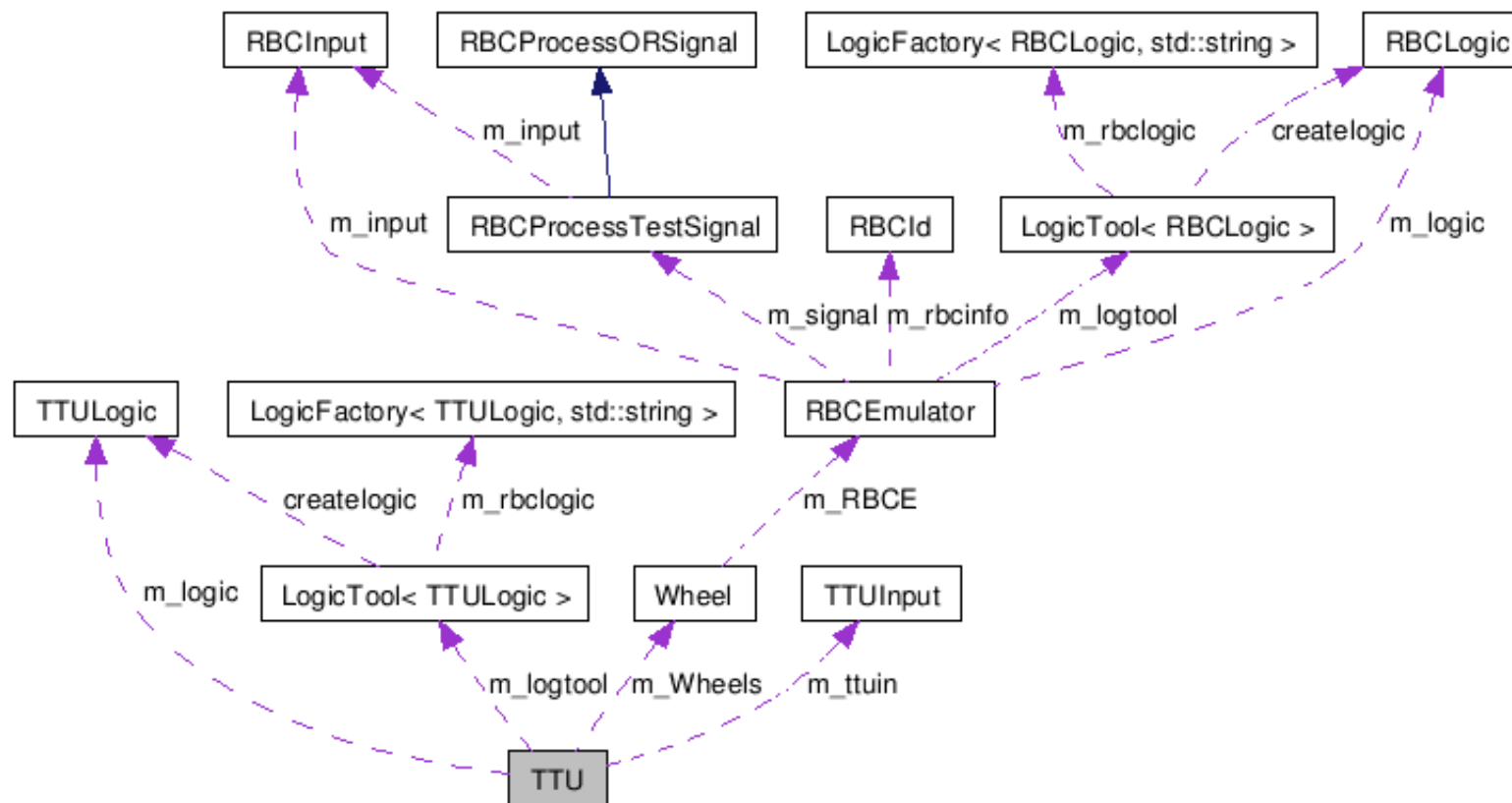
Class collaboration diagram (using Doxygen): RBCEmulator



# Implementation: Class Collaboration



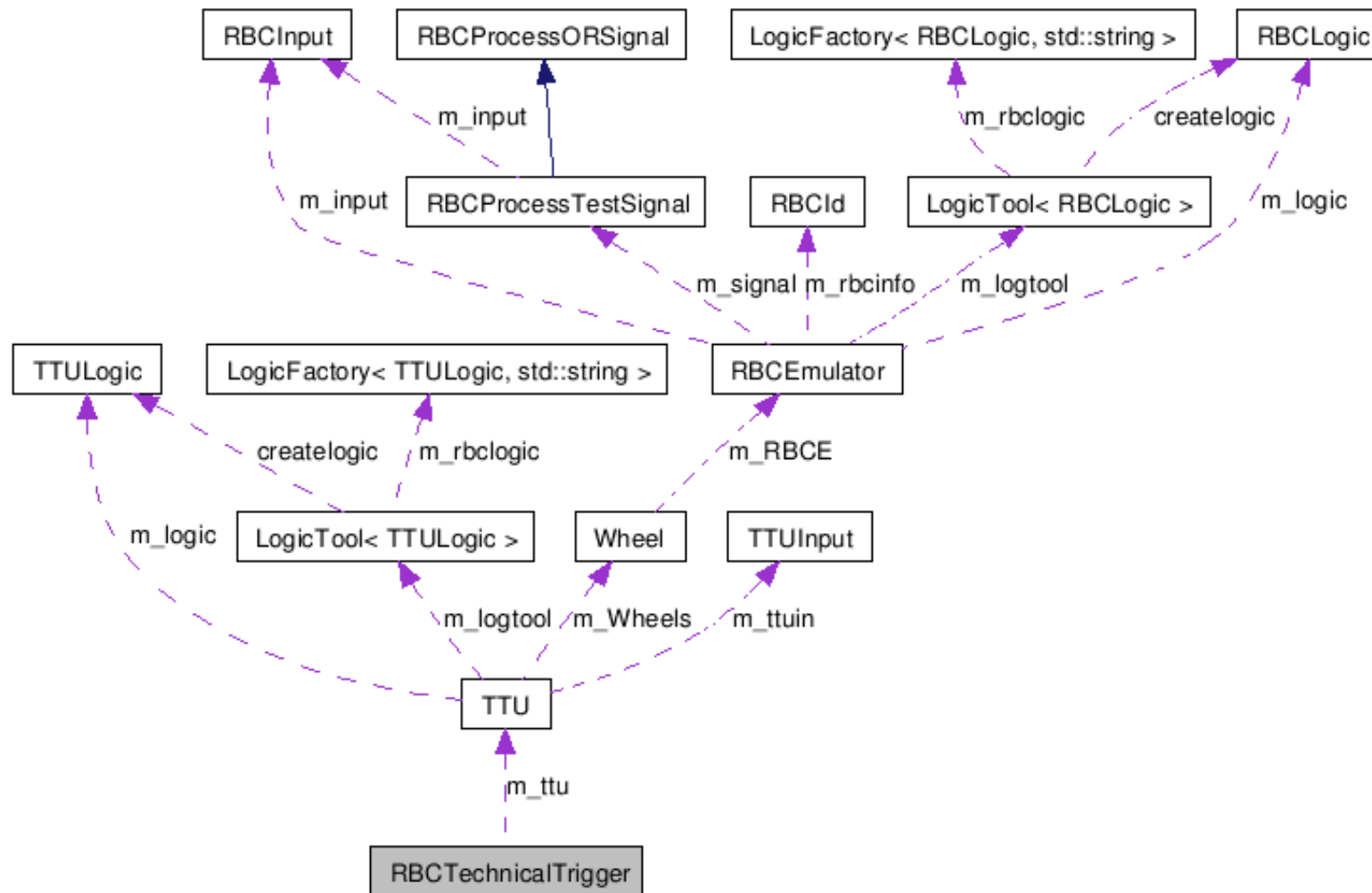
Class collaboration diagram (using Doxygen): TTU



# Implementation: Class Collaboration



Class collaboration diagram (using Doxygen): RBCTechnicalTrigger



- The emulator design is in place: the infrastructure is up and running:  
from local (RBC boards) → to global (TTU) and RBCTT
- Two kinds of logic are written at the moment:
  - RBC Level: Coincidence Logic
  - TTU Level: Tracking Algorithm
- I have the mapping between 26/28 ORs to 6 layers per sector
- From this information and using a coincidence logic, I can obtain a wheel map
- You can access the code from our SVN repository:  
`svn checkout`  
`svn+ssh://svn.cern.ch/repos/cmsuniandes/Users/aosorio/Code/RPCTTindep`
- There is a README file that explains how to compile and run the test

Work to to be done:

- Of course, design and implementation is not necessarily perfect: need to keep iterating
- I am testing the Tracking Algorithm
- At the last MuonDPG/RPC group meeting (21/10), Marcello and Marcin mentioned the integration of the code in some Emulator Framework → need to investigate this!
- Presentation in 2 weeks time

## References

- [1] Loddo F. *An RBC based Technical Trigger for the CMS Experiment*. xxxxx, Valencia, 2007.