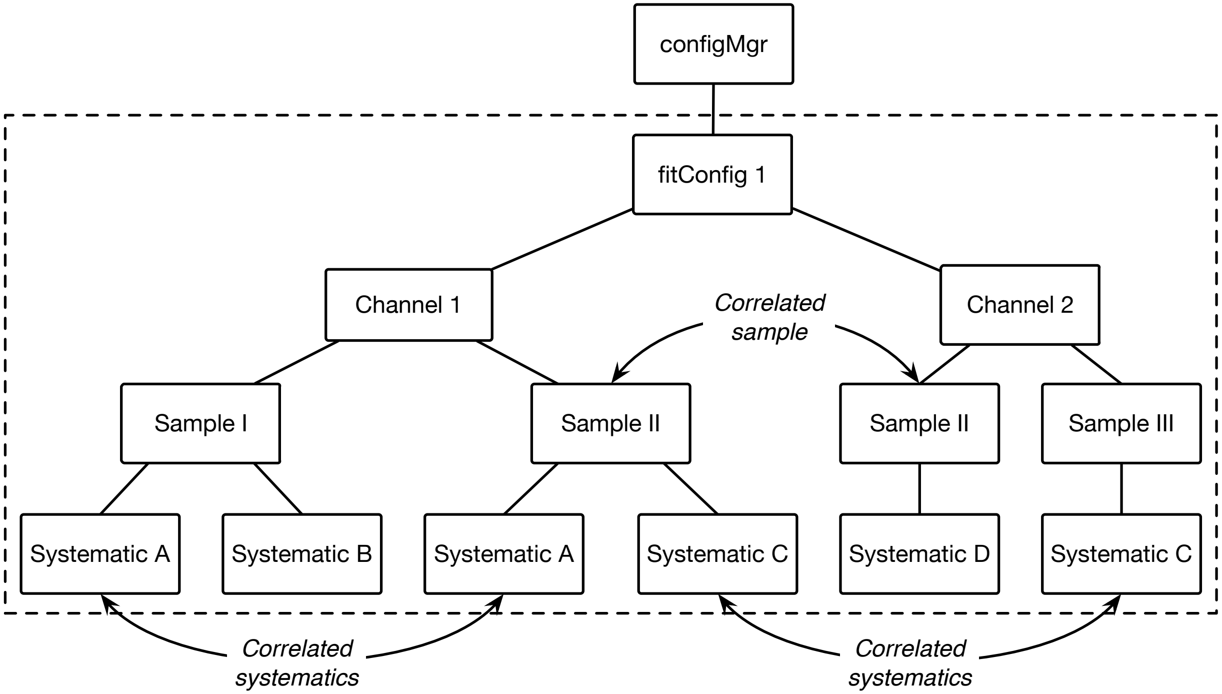
HISTFITTER CONFIGURATION FILE

K. Hamano

Configuration example



- Channel: SR, CR1, CR2, VR1, VR2, etc.
- Sample: Signal, ZZ, WG, Z+jets, etc.
- Systematic: Electron ID, Jet energy scale, etc.

Key words in the configuration file

- HistFitter use fitConfig class to construct PDF.

```
from configManager import configMgr
myFitConfig = configMgr.addFitConfig("myAnalysisName")
```

- Adding channels:

```
myChannel = myFitConfig.addChannel("myObs", ["mySelection"], nBins, varLow, varHigh)
myUnbinnedChannel = myFitConfig.addChannel("cuts", ["mySelection"], 1, 0.5, 1.5)
```

  - To specify the channel is CR, VR or SR:

```
myFitConfig.setBkgConstrainChannels(myChannel)
myFitConfig.setValidationChannels(myChannel)
myFitConfig.setSignalChannels(myChannel)
```

- Adding samples:

```
mySample = Sample("SampleName",myColor)
myChannel.addSample(mySample)
```

  - Input to the samples can be, TTree, Float or Histogram.

Key words in the configuration file

- In case of TTree or Float, the HistFitter creates histograms from those
    - Float example: 3 bins with events 100, 34 and 220

```
mySample.buildHisto([100,34,220], "region", "observable")
```

    - TTree example:
        - SampleName = tree name.

- Adding Systematics:
    - Define systematics:

```
mySys = Systematic("myTreeSys", "ASample", "ASample\_UP", "ASample\_DOWN", "tree", "myMethods")
mySys = Systematic("myWeightSys", ["nominalWeights"], ["upWeights"], ["downWeights"], "weight", "myMethods")
mySys = Systematic("myUserSys", ["nominalWeights"], 1.1, 0.8, "user", "myMethods")
```

    - Add to either sample or channel

```
mySample.addSystematic(mySys)
myChannel.addSystematic(mySys)
```

        - Sample -> specific to the sample.
        - Channel -> common to all samples in the channel.