Miscellaneous thoughts on future developments

E. Rodrigues
University of Manchester

LHCb B2OC time-dependent Workshop , Padova, 9-10 July 2015

Random thoughts & some code for discussion …

Use, reuse … and abuse !

- Usual statement : do not reinvent the wheel !
- Recent developments of the package go very much in this direction – great
- YES, just an acknowledgment to Manuel and Agnieszka ☺ !

Documentation

- ❑ Everyone wants it …
  but nobody wants to write it :S !

- ❑ I do have a strong opinion that it is very important
- ❑ I won't even argue here about its need

- ❑ Do we prefer/want Pythonic style of options?
  Or the Google style?
- ❑ What's the direction of the community these days?

- ❑ One should also think in terms of what gives us nicer browsable doc
  such as the one provided by Doxygen

B2DXFitters – possible future directions

- What's D and X in B2DXFitters ?
- As the code and package becomes more versatile and comprehensive, does it still make sense to call it B2DXFitters ?
- The project is being repackaged to reflect core and "universal" code from code specific to particular analyses / decay modes / etc.
- This should be done asap, i.e. now …

B2DXFitters – now going a bit wild …

- Time-dependent amplitude analyses will start in LHCb at some point …
- Worth keeping that in mind when developing the project !
- Many customers outside the B2OC WG

Model builders to stick to common conventions

```
model = GaussianPDFBuilder( ws,
                            'Bu2KSPi',
                            mass,
                            { 'mean'  : 5300.,
                              'sigma' : 20.,
                              'Debug' : True
                              }
                            )

model = DoubleCrystalBallPDFBuilder( ws,
                                     'Bs2DsPi',
                                     mass,
                                     { 'mean'    : 5280.,
                                       'sigma'   : 20.,
                                       'alpha1'  : 2.,
                                       'n1'      : 1.5,
                                       'alpha2'  : 1.,
                                       'n2'      : 1.,
                                       'frac'    : 0.6,
                                       'Extended' : True,
                                       'events'  : 500,
                                       'Debug'    : True
                                       }
                                     )
```

One of "helper functions",
or wrapper if you want

Model builders – calling in a different way

- ❑ "All" fit model helper functions in module fitmodulebuilders.py
  - Makes sure all code follow the same conventions

```
// Calling the model builders by name
model = getattr( fitmodelbuilders,
                 fitmodelbuilders.modelbuilders[modelName] )(
                   ws,
                   decayName,
                   mass,
                   config['ModelDefaultParams'][modelName]
                   )
model.Print('t')
```

Model builders – fit model configuration via Python dict

```
// Configuration file for a multi-PDF builder
// ...
# Modes to include in the fit model
  'Modes' : [ 'Bu2KSPi',
              'Bu2KSK',
              'B2KSHH',
              'CombBkg'
              ],

  # Models for each fit component
  'Models' : { 'Bu2KSPi' : DoubleCrystalBallPDFBuilder,
               'Bu2KSK'  : DoubleCrystalBallPDFBuilder,
               'B2KSHH'  : PDFFromFile,
               'CombBkg' : ExponentialPDFBuilder
               },

  # Model parameters
  'ModelParams' : { '2011LL' : modelParamsConfig2011LL ,
                    '2012LL' : modelParamsConfig2012LL ,
                    '2012DD' : modelParamsConfig2012DD
                    },
// ...
```

Model builders – generic generation of fit model

```
 model = MultiEPDFBuilder( ws,
                             `MyFavouriteModel',
                             mass,
                             config[ 'Modes' ],
                             config[ 'Models' ],
                             config[ 'ModelParams' ][ sample ],
                             True
                             )
```

P.S.: Yes, that's it ;-)

P.S.2: OK, does not cover all use cases, but probably 90% of them !

Model builders – generic generation of fit model

```
// Configuration file for a multi-PDF builder
// ...
modelParamsConfig2011LL =  { 'Bu2KSPi' : { 'mean'        : 5280.,
                                            'sigma'       : 1.7120e+01,
                                            'alpha1'      : 1.6898e+00,
                                            'n1'          : 1.8130e+00,
                                            'alpha2'      : -1.3127e+00,
                                            'n2'          : 3.4685e+00,
                                            'frac'        : 7.8017e-01,
                                            'Extended'    : True,
                                            'ConstParams' : [ 'sigma',
                 'alpha1', 'n1', 'alpha2', 'n2', 'frac' ],
                                            'events'      : 600,
                                            'Debug'       : True
                                            },
// ...
```

Random utils – saving in toy studies

```
// Extending the idea of configurations via dictionaries
saveConfig = { 'Workspace' : pdfDict[ 'ws' ],
               'FitResult' : ( result if fitConfig[ 'SaveFitResult' ]
                                 else None ),
               'OutputDir' : fitConfig[ 'OutputDir' ],
               'Prefix'    : 'Bu2KSH',
               'IsToy'     : fitConfig[ 'IsToy' ],
               'ToyNumber' : toyNumber,
               'Debug'     : True
             }
saveObjsToFiles( saveConfig )
```

- **It saves typically 2 files separately for the fit result and the complete model**
  - For toys: <Prefix>_fitresult_toy-<%04d>.root & <Prefix>_modelWS_toy-<%04d>.root
  - For data: <Prefix>_fitresult_data.root & <Prefix>_modelWS_data.root

- **One can also save any other set of objects to a separate file with a dedicated dict key**