Subject: Data levels in R3Broot - suggestion
Posted by Ralf Plag on Wed, 20 Jan 2016 12:07:48 GMT
View Forum Message <> Reply to Message

To avoid further chaos we need to define some common data levels.

In land02 we have:

RAW-level: mapped data (e.g. plane, bar, pm), no calibration applied
TCAL-level: time-slope and energy-offset applied
SYNC-level: fully calibrated: time-sync and energy-slope applied as well
DHIT-level: Physical quantities in the detector internal coordinate system
HIT-level: Physical quantities in cm, ns, MeV

In ucesb we have:
UNPACK: module, channel
RAW: mapped, as above

Unfortunately, in R3BRoot, RAW-level corresponds to the ucesb UNPACK-level.

I'd like to suggest three base levels which every detector should use. Additional levels can be added if needed.

MAPPED-level: mapped data (e.g. plane, bar, pm), no calibration applied
CAL-level: Fully calibrated
HIT-level: Physical quantities in cm, ns, MeV

Note:
UNPACK level is not needed in R3BRoot since this is provided by ucesb in stand-alone mode.
Since the term "RAW-level" is ambiguous, I suggest not to use it at all. MAPPED is nicely descriptive and intuitive, so we better use that and it should always refer to the detector channel (e.g. bar, pm) and not to the electronics channel (module, channel).
Existing data structures called "RAW" (mostly holding UNPACK-data) should be deleted (or if needed for some strange reason renamed into UNPACK).

Cheers,
Ralf

Subject: Re: Data levels in R3Broot - suggestion
Posted by Dmytro Kresan on Thu, 21 Jan 2016 10:10:00 GMT
View Forum Message <> Reply to Message

Hi Ralf,

I fully agree to the scheme you have proposed.

We will not need "old" RawItem classes as soon as R3BRoot unpackers (3 or 4 existing) will be substituted by Ucesb Readers. But I would like to keep them for the transition period, so I will rename them to Unpack level.

Technical question, about the naming convention. For example, R3BNeulandHit and R3BNeulandDHit are self-informative. Do we need "Item" suffix for Mapped and Cal levels, something like R3BNeulandCalItem?

Cheers,
Dima

---

Subject: Re: Data levels in R3Broot - suggestion
Posted by Ralf Plag on Thu, 21 Jan 2016 13:15:38 GMT
View Forum Message <> Reply to Message

Hm...
Besides the classes that hold the data, there are also classes that convert the data from one level to another. If we always keep the "Item" for the data classes, we could use the version without "Item" for the conversion classes, that convert into the named level:

data-class: R3B[detector][level]Item
conversion-class: R3B[detector][level]

E.g.: class R3BLandCal would convert data and write it into R3BLandCalItem

Alternatively, we could have another naming scheme for the conversion classes like:

R3BLandRaw2Cal or R3BLandFillCal or ...?

---

Subject: Re: Data levels in R3Broot - suggestion
Posted by Ralf Plag on Thu, 21 Jan 2016 13:19:26 GMT
View Forum Message <> Reply to Message

Appendix:

Currently there is for the conversion classes:

R3B[detector][level]Ana

but I'm always unsure whether level corresponds to the input or output data of the conversion.

---

Subject: Re: Data levels in R3Broot - suggestion
Posted by Dmytro Kresan on Thu, 21 Jan 2016 14:23:47 GMT
View Forum Message <> Reply to Message

Ralf Plag wrote on Thu, 21 January 2016 14:15

data-class: R3B[detector][level]Item
conversion-class: R3B[detector][level]

E.g.: class R3BLandCal would convert data and write it into R3BLandCalItem

Fully support. This will provide descriptive matching of data and tasks.

Concerning input and output: I would stick to the naming of conversion classes according to the output it produces. Since this is what task supposed to do.

An issue remains: what about 2 stages of TCAL calibration: one for calculating the parameters and another for data conversion? How can we name the first one? R3BNeulandCalCollect, R3BNeulandCalFill?

---

## Subject: Re: Data levels in R3Broot - suggestion
## Posted by Hector Alvarez Pol on Tue, 02 Feb 2016 18:34:43 GMT

View Forum Message <> Reply to Message

Hi,

if it's not too late for the definition of the levels, I would like to comment on this topic:

- I agree with the Mapped, Cal, Hit as expressed by Ralf, but this is not a complete list. There could be intermediate calibration levels and different complexity of Hit structures. So, we should give space for Mapped, PreCal1, PreCal2, ..., Cal, Hit, Cluster, Track, ... in principle for a detector, but maybe using information of two or more detectors for the more complex (or complete) levels. We could give a list preferred name levels (fixing some standards as Mapped, Cal and Hit) and adapt and guide the developers for intermediate levels.
- I would not use the word Item for data levels. Just the name composed of R3BDetectorLevel (as in R3BLandCal). For me it should clear enough and identify it .
- I would use instead a modifier for the classes converting data levels. The class to obtain a given level would be R3BDetectorLevelFinder. Another synonym for Finder could also make the job, if you find any better. The second proposal by Ralf (R3BDetectorLevelFrom2LevelTo) is also valid for me, and maybe could work better if there are alternative tasks with different possible origin or destination levels (I do not know if this could happen). But I would prefer the first method according to Dmytro comment ("I would stick to the naming of conversion classes according to the output it produces. Since this is what task supposed to do.")
- Regarding parameters, I would use R3BDetectorLevelPar to identify the parameters used to convert data to a given Level.
- For tasks calculating or collecting parameters (geometrical parameters, calibration parameters, tracking parameters, reconstruction parameters, matching windows, ...) I would use a clear R3BDetectorLevelParFinder.

Regards,

Héctor

---

Subject: Re: Data levels in R3Broot - suggestion
Posted by Dmytro Kresan on Wed, 03 Feb 2016 07:16:31 GMT
View Forum Message <> Reply to Message

Hector Alvarez Pol wrote on Tue, 02 February 2016 19:34
- I agree with the Mapped, Cal, Hit as expressed by Ralf, but this is not a complete list. There could be intermediate calibration levels and different complexity of Hit structures. So, we should give space for Mapped, PreCal1, PreCal2, ..., Cal, Hit, Cluster, Track

Ralf has mentioned about introducing intermediate levels on demand. This option is definitely not excluded.

Hector Alvarez Pol wrote on Tue, 02 February 2016 19:34
- I would not use the word Item for data levels. Just the name composed of R3BDetectorLevel (as in R3BLandCal). For me it should clear enough and identify it .

This will work better, since names of data classes are used more often in the analysis code than names of tasks.

Hector Alvarez Pol wrote on Tue, 02 February 2016 19:34
- I would use instead a modifier for the classes converting data levels. The class to obtain a given level would be R3BDetectorLevelFinder. Another synonym for Finder could also make the job, if you find any better.

To my opinion - "Finder" is in some cases misleading. It is suitable for clusters reconstruction, hit finding, etc, but not for calibration.

Subject: Re: Data levels in R3Broot - suggestion
Posted by Ralf Plag on Wed, 03 Feb 2016 09:29:56 GMT
View Forum Message <> Reply to Message

Hi All,

I want to strongly encourage using descriptive names. This makes life easier, especially for beginners, and we need as many people (aka beginners) as possible to work on R3BRoot.

And that's why I like the suffix "Item": It describes that the class holds ONLY ONE Item, as opposed of all MAPPED or CAL level data.
...and why I like Hector's suggestion to add an suffix to the converter classes. What about "Builder"? E.g. R3BTofHitBuilder?
...and why I don't like the names of the suggested intermediate levels precal1 and precal2 (or have that been only generic examples?) What is the difference between those two and between Cal?

And as already mentioned: Of course we need also other levels than MAPPED, CAL, HIT. My idea was the these three levels should be available for ALL detectors. Other levels can be always introduced in addition, as needed, although it would be fine if we could define at least some common names for those additional levels as well.

However we decide, we should decide fast since a lot of development is already ongoing. I'd like to set end of next week as deadline for the naming decision.

Cheers,
Ralf

---

Subject: Re: Data levels in R3Broot - suggestion
Posted by Hector Alvarez Pol on Wed, 03 Feb 2016 10:14:38 GMT
View Forum Message <> Reply to Message

Hi,

yes, we should converge soon. And I agree that descriptive names are the best for everyone.

- I also agree with Builder as Ralf said. As pointed out by Dmytro, Finder is not suitable for all, but it could even be better to simply say that it could be Builder or any other similar word that suits perfectly to the task (as could be "Converter", "Filter", "Calibrator"???).
- Precal1, Precal2 were simply generics. I was trying to state that we can have several intermediate data levels. I agree with you that intermediate levels could be added for detectors provided the name is clear enough, and we should fix some basis (Mapped, Cal, Hit, ...). We could ask simply for agreement before selecting the name of additional intermediate states to ensure that the name is ok.
- The only point where we disagree (I think) is in the Item for data levels. In my opinion, a level like R3BNeuLandHit or R3BCalifaCrystalCal is clear enough, and I do not need the word Item to understand that is a data level. It is impossible to mistake them with "conversors" as they will have the Builder suffix, or parameters with the Par suffix. So, I do not understand Ralf's quote: "It describes that the class holds ONLY ONE Item, as opposed of all MAPPED or CAL level data", because all classes are abstraction of objects and are a single item in a sense. Then, one could instantiate many objects... Maybe I did not get the point properly.

Best regards,

Héctor

---

Subject: Re: Data levels in R3Broot - suggestion
Posted by Ralf Plag on Wed, 03 Feb 2016 12:21:42 GMT
View Forum Message <> Reply to Message

Hi Hector and All,

I do understand what you mean. However, I'm trying to look at the project from an outsider's perspective. Meaning: I'm trying to "forget" everything I know about R3BRoot. And then I wouldn't have any idea what R3BCalifaCrystalCal is about. I probably would assume it is performing some calibration.

We just had a small internal discussion (Dima, Haik, me) and came to the following suggestions:

Converters should be called R3B[Detector][FromLevel][ToLevel], e.g.
R3BCalifaCrystalMappedCal or maybe with a '2': R3BCalifaCrystalMapped2Cal

Data classes should be called R3B[Detector][Level]Data

Calibration parameters should be called R3B[Detector][Level]Par

I admit that the 'Data' is not strictly necessary but it nicely marks all classes of the same type, which makes reading code for newbies much easier.

We also decided to collect more ideas here up to Friday this week and then send around the final/best idea via email to the analysis mailing list to give everybody else the chance to raise their voice.

Cheers,
Ralf

---

Subject: Re: Data levels in R3Broot - suggestion
Posted by Hector Alvarez Pol on Wed, 03 Feb 2016 12:33:09 GMT
View Forum Message <> Reply to Message

Hi,

ok, I agree with your suggestion.  I would prefer the 2 between the levels in the converters (as in R3BCalifaCrystalMapped2Cal).

Regards!

Héctor

---

Subject: Re: Data levels in R3Broot - suggestion
Posted by Pablo Cabanelas on Wed, 03 Feb 2016 13:40:12 GMT
View Forum Message <> Reply to Message

Hi all:

I do also agree with the convention, and I would prefer a "2" too

Tech. question: would it be advisable to rename the TClonesArray collection as well? e.g. TOFPoint --> TOFPointData

---

Subject: Re: Data levels in R3Broot - suggestion
Posted by Dmytro Kresan on Thu, 04 Feb 2016 09:48:27 GMT
View Forum Message <> Reply to Message

I propose to have a branch name in the output tree same as the data class name, but without prefix R3B. In this case it will be easy to recognise type of object stored. For example class R3BNeulandCalData will be stored in a branch "NeulandCalData". Concerning simulation part, I would not touch it now. So class R3BTofPoint is stored as TofPoint. The only change in the simulation part we have to do now, is that Digitizers or HitFinder arrive at the same HIT level as calibrated and analysed data.

Cheers,
Dima