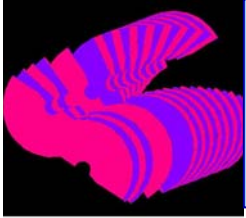


Alignment Framework and VELO SW

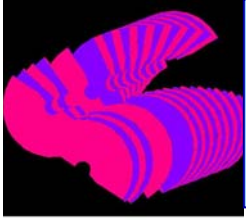
Alignment framework status
Changes to VELO geometry
Required changes to VELO software

Juan Palacios
CERN

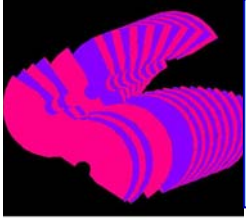


Alignment framework status

- **Possible to apply misalignments to detector elements in detector data service**
 - Stored as XML strings either in XML file or conditions database
- **Can be used wherever the DetDataSvc and detector elements are used**
 - Visualisation, digitisation, reconstruction: OK
 - Simulation: **some issues still to be resolved...**

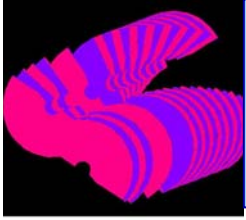


- **Functionality requires changes from potential users**
 - XML description must point to conditions
 - All client software must make good use of them!
- **XML part has been taken care of. This has highlighted some weaknesses in the code that should be addressed.**



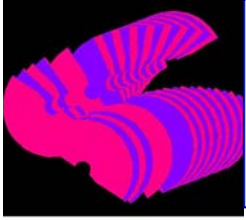
Applying misalignments: XML

- **XML is used for various things**
 - Defining Detector elements
 - Defining catalogues
 - Structure (detector elements)
 - Geometry (definitions of volumes, used by detector elements)
 - Conditions
 - Defining conditions
 - At the moment, alignment conditions are six parameters encapsulated in an XML string
 - These can be stored in a file or in a data base



Applying misalignments: XML

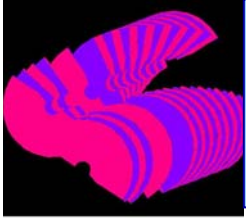
- **Each object that you want to align independently has to be a `DetectorElement`**
 - This is done in the XML by assigning logical volumes to detector elements.
- **Each detector element must contain a path to an “`AlignmentCondition`”**
- **These should resolve to a folder on `CondDB` or a place in an XML catalogue**



Applying misalignments in VELO XML

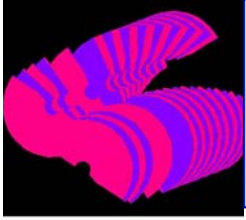
- **This has already been done in latest Det/XmIDDDDB**
 - VELO re-structured to allow to misalign various parts (details later)
 - Conditions assigned to each detector element
 - Catalogue of dummy conditions created
 - Test CondDB filled with “snail” parameters

See [Det/XmIDDDDB/v26r0/DDDB/Velo/v200507](#)



Applying misalignments in VELO XML

- **Users should use and follow examples in new Det/XMLConditions package**
 - Contains conditions catalogues for local XML file or test CondDB conditions
 - Use only local copy as only I can access the data base at the moment
 - Allows to change alignment parameters without having a copy of the whole XmlDDDB
 - Allows to switch between XML file and real data base conditions



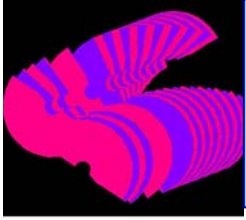
Old vs. New XML det. elem. structure

DeVeloSensors: /dd/Structure/LHCb/Velo/DetectorXX



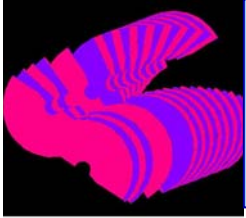
DeVelo: /dd/Structure/LHCb/Velo

Old VELO



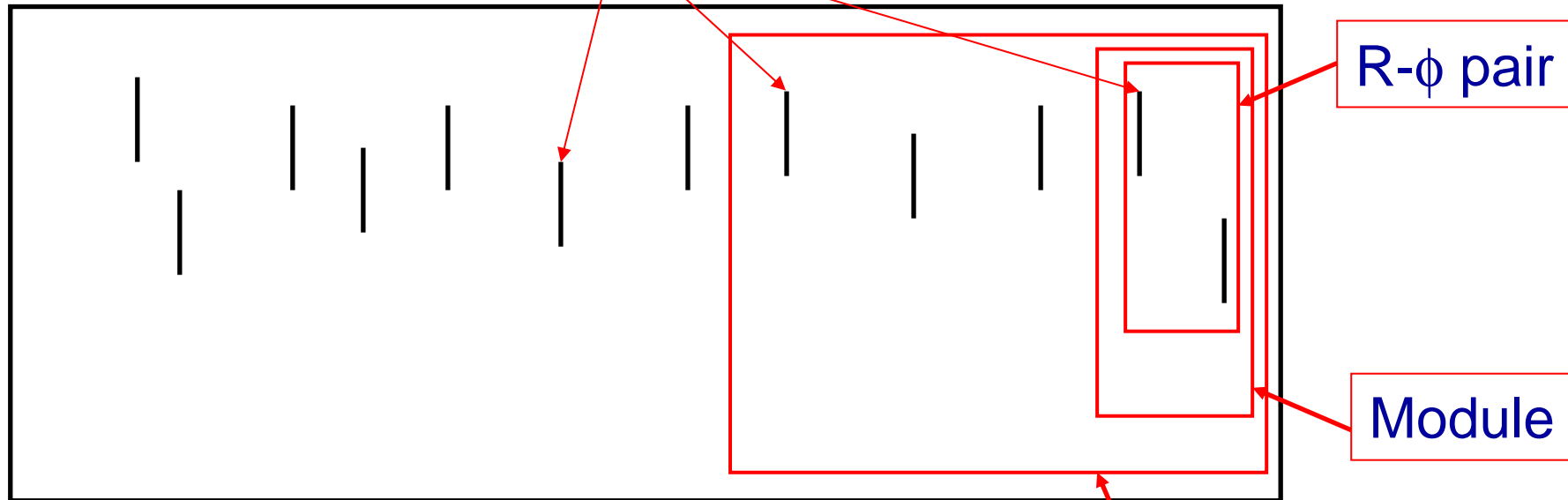
Problems with old VELO XML

- **Correlated movements of sensors tricky**
- **Geometry structure was actually different to det. elem structure!**
 - Previous misalignment studies act on geometry, ie move sensor pairs, but these were not accessible as detector elements...
- **Specialised VELO det. elem. Code written to work only with this description**
 - Designed to fall over with the smallest of changes
 - Did not permit to have detector elements other than sensors inside VELO



Old vs. New XML det. elem. structure

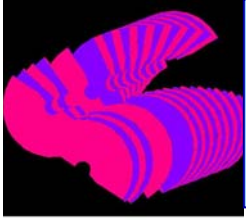
DeVeloSensors: /dd/Structure/LHCb/Velo/DetectorXX



DeVelo: /dd/Structure/LHCb/Velo

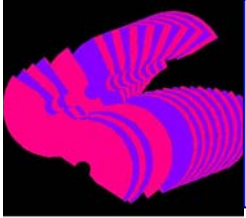
VeloHalf

New VELO



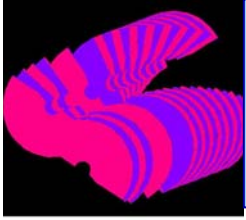
XML summary

- **New structure allows to misalign at many levels:**
 - Whole VELO
 - Each half
 - Each module
 - Each sensor
- **Specialised DeVelo code changed to find the DeVeloSensors inside DeVelo**
- **Bonus: each sensor now an individual logical volume: can have **different thicknesses** for each one!**



XML to-do

- **New description only places modules in VELO**
 - Need RF foil and boxes, vacuum tank, RF wakefield guide cone...
 - These can be incorporated in a way which will allow simulating a full VELO with misaligned halves or modules (but not sensors)
 - Review r - ϕ separation, hybrid material, positions of modules
- **M. Tobin doing all this *as we speak!***

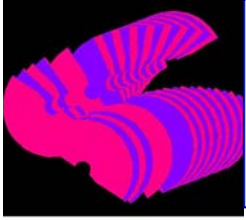


Using misalignments

- **Old detector element interface returns pointer to geometrical information**
 - This now contains ideal+misaligned transformations
 - Using a misaligned detector is transparent to the user.
- **The misalignment matrix can be obtained and updated**
 - `detElem->geometry()->alignmentCondition()`

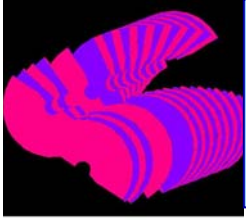
**Can pass it a new matrix or parameters.
Will trigger re-calculation of local-global transf. matrices
in self and daughter detector elements**

http://lhcb-release-area.web.cern.ch/LHCB-release-area/LHCB/doc/html/class_alignment_condition.html



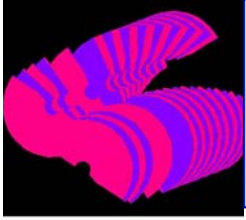
Making misalignments

- **AlignmentConditions are created from XML strings by converters**
 - Requires presence of either
 - An XML file with the parameter definitions and values
 - An XML string in the database
- **Can make manually**
 - From HepTransform3D or vectors with parameters
- **Can turn into XML string and upload to CondDB**
 - `AlignmentCondition::toXml()`



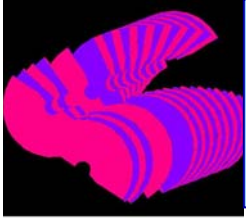
VELO Code changes

- **Existing code:**
 - Robustness
 - Caching of information
 - Use of UpdateManager
 - Use of XmlConditions package
- **New code:**
 - VELO Alignment using new DetDesc
 - Populating CondDB. See Ex/DetCondExample
 - Tracking with misalignments



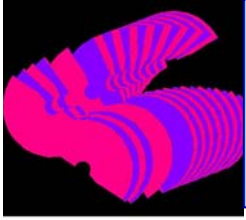
Changes to existing code

- **Problems came to light through this exercise**
 - Fragile code, reliant on very special geometry configuration
 - DeVelo largely culpable, but client code also makes stupid assumptions sometimes
- **Maybe it is a good moment to fix a few things**



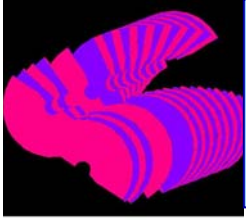
Existing code maintenance

- **Identify where geometry or structure assumptions are made and their impact**
 - Hard-wired Z windows to identify sensors
 - Reliance on array indices
 - Assumptions of perfect alignment in digitisation/reconstruction
 - Reliance on sensor numbers: safer to have numbering convention-free software
- **Who is responsible for what bits of VELO code?**
 - Must know who to address problems and bug reports to



New code

- **Fortunately I have already made changes to VeloDet to make all of this easy...**
- **Alignment inside framework**
- **Populating CondDB with arbitrary alignment parameters**
- **Tracking with displaced sensors**
 - Is it really true that there is **no intention of doing this?**
 - If so, why bother to align the detector, and to have alignment parameters in the conditions database?
 - Why even have an alignment framework?
 - I hope this is a misunderstanding!



Conclusions

- **Geometry framework allows easy use of detector misalignments**
- **The VELO has been used to test this**
 - Many software changes have already been made
 - You are in an advantageous position to start using this!
- **Issues to be resolved:**
 - XML and DeVelo caching: Mark working on it
 - Simulation of misaligned detector: I am working on it, but Mark's XML provides a temporary fix for VELO
 - Writing to the actual data base: should sort out once you have code that can generate conditions and needs to store them!
 - Requires an authorised user and a clear policy