
Packaging fullsim software

Andrea Di Simone
INFN Tor Vergata

Proposals

- After some discussion on the fullsim mailing list, several proposals were done on how to split the fullsim code into packages
- General agreement to distinguish between “Core” and “Detector” code
- Within those domains, the level of splitting depends on the actual amount of code involved
 - Avoid having more packages than files...
- Most likely, will start with a minimal set of packages, aiming at having a more complex structure in the longer term, to reflect the growing complexity of the simulation code

The geometry problem

- Aim is to also split the gdml description for the different subdetectors
 - In practice, this means that each subdetector will manage its own geometry description
 - Keep in “Core” only top volumes, to better handle space allocation and volume clashes
- The present structure is already highly modular, but requires the full path to all the gdml fragments to be known
 - No entity resolving based on search path foreseen
 - Not good for use of distributed computing resources
 - This limitation comes from xerces-c, not from G4 itself
- I don't see this as a show stopper, since one could find workarounds
- The easiest solution would probably be to prepare a small script
 - Do filename resolution based on search path
 - Symlink resolved physical files in local run dir
- Sooner or later, xerces-c will provide support for “clever” name resolving, for example using xmlcatalog

Other issues

- Whatever packaging structure is chosen, several modifications will have to be done to fullsim
 - C++ side (ADS + EP + ???)
 - Build procedure (???)
 - Repository structure (RS?)
 - Releasing and distributions (RS?)
 - Validation (???)
- Main issue (imvho) will be manpower
- Of course, any involvement of present human resources in this packaging will disrupt normal development and user support
 - Involvement of subdetectors experts is crucial