# *"Educated guess physics lists for geant4 hadronic physics"*

J.P. Wellisch
CERN/EP/SFT

# *What does it need to make an "educated guess"*

- Sufficient richness in modeling
- A user base, that is willing to iron things out (I.e. beta testers)
- Experience in the various areas of application.
- Sample use-case realizations

- … Funding … support … time … effort …

# *The starting point*

- It is well recognized since quite some time that writing a good physics list is no trivial, in particular when hadronic physics is involved.
- It is nice to be able to exploit the full power in the flexibility and variety of hadronic physics modeling in geant4, but being forced to do so is not what we want.
- It is also nice to have the physics transparently in front of you and be able to exploit it in the best possible way, but being forced to understand it all is (very understandably) not what people want, either.

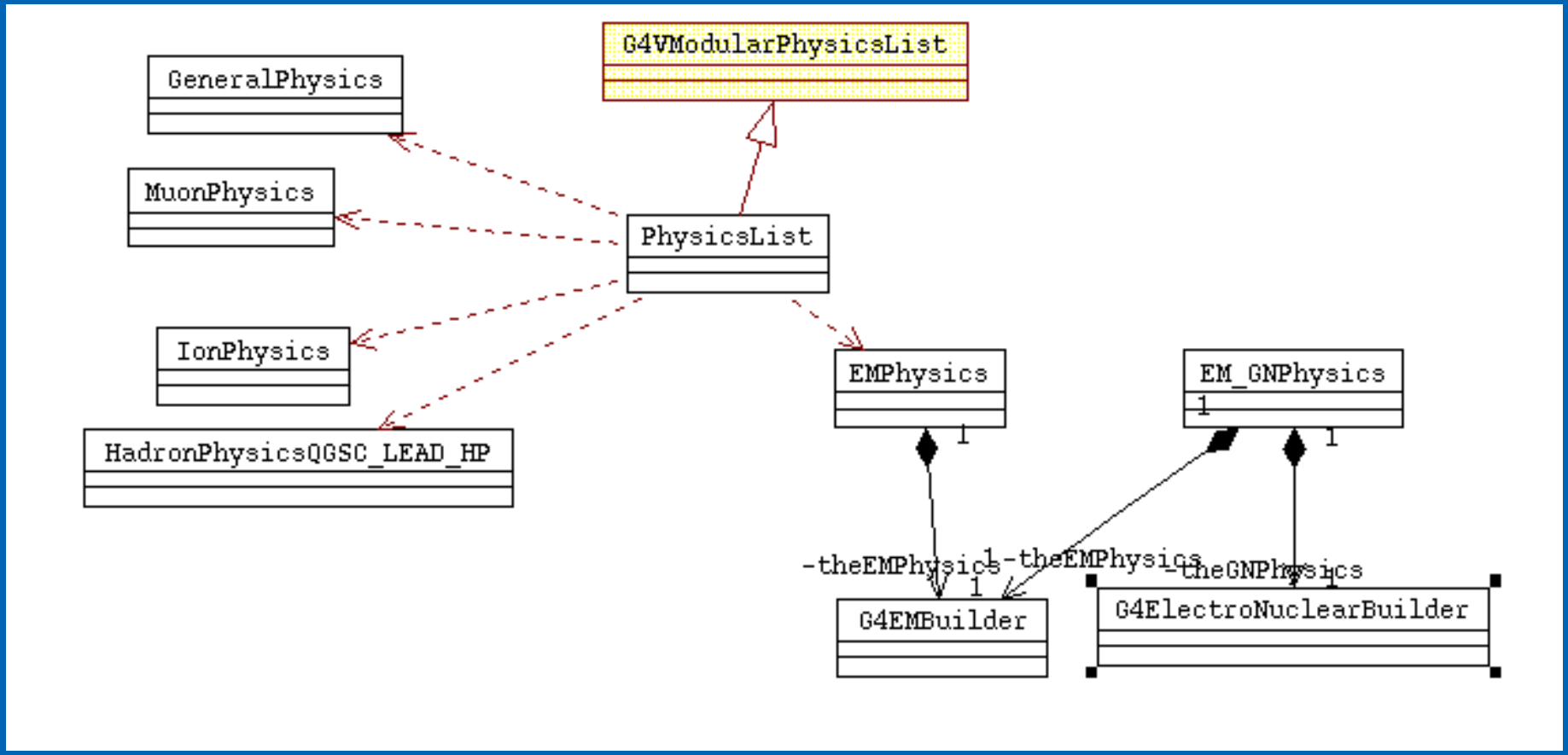
# *Because of this*

- We have systematically accumulated experience with various combinations of cross-section and models over the last years.
- Today we can provide a set of physics lists institutionalizing this experience.
- Publishing them to the general audience was one of the main milestones of the hadronic working group for 2002.

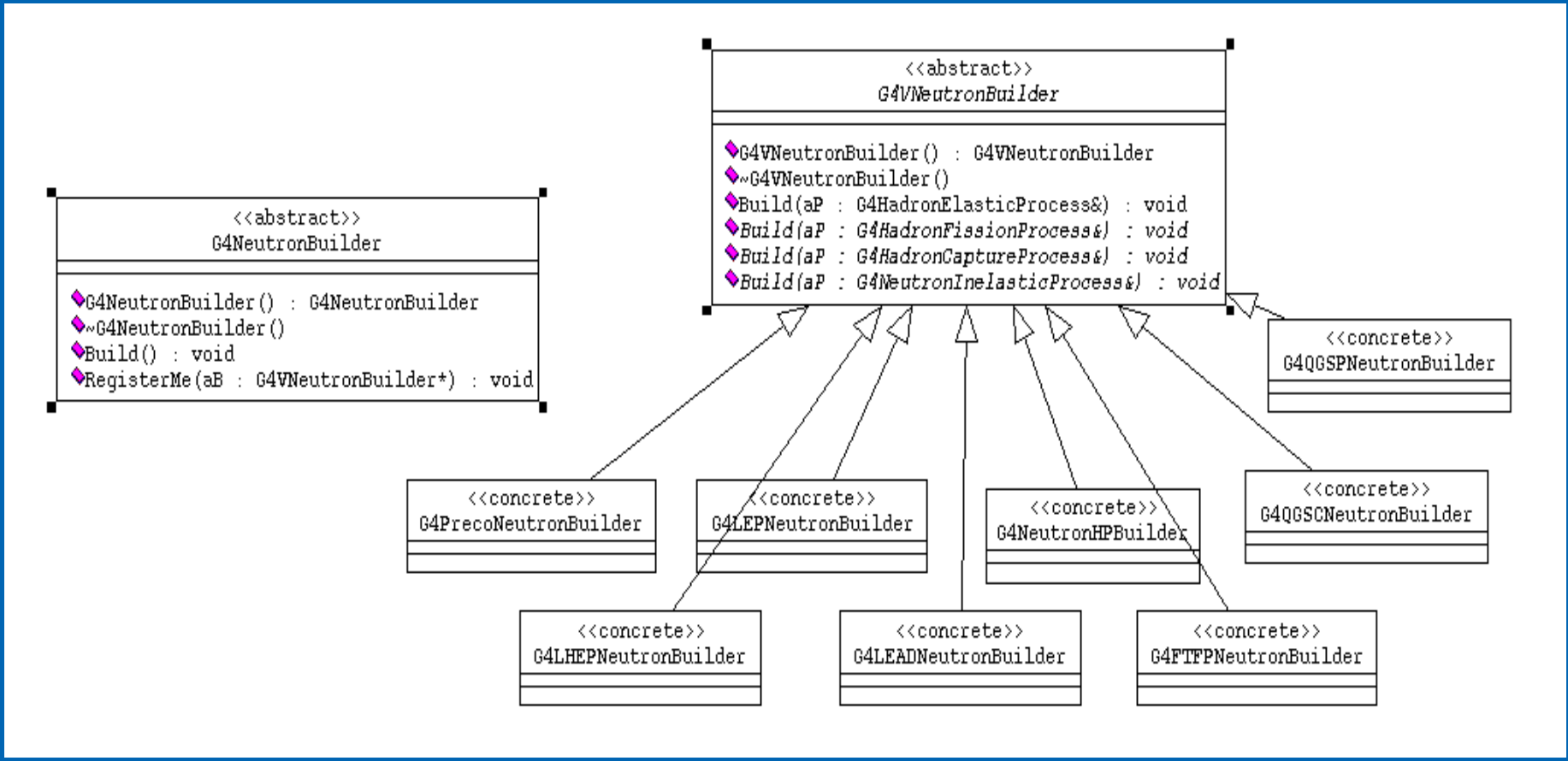
# *Mission statements*

- Give an 'educated guess' that can be used as starting point by any user and/or give meaningful alternatives for physics lists for the various use-cases of hadronic physics.
- Provide a brief description of the modeling used in the different physics lists.
- Find a natural organization for the verification/validation results, so people can find the part of the information they are interested in; whether they do calorimeter design or neutron dosimetry.
- Provide for areas where users can contribute their findings.
- Provide a distribution mechanism for both physics lists and verification results.
- Make physics lists source code more readable.

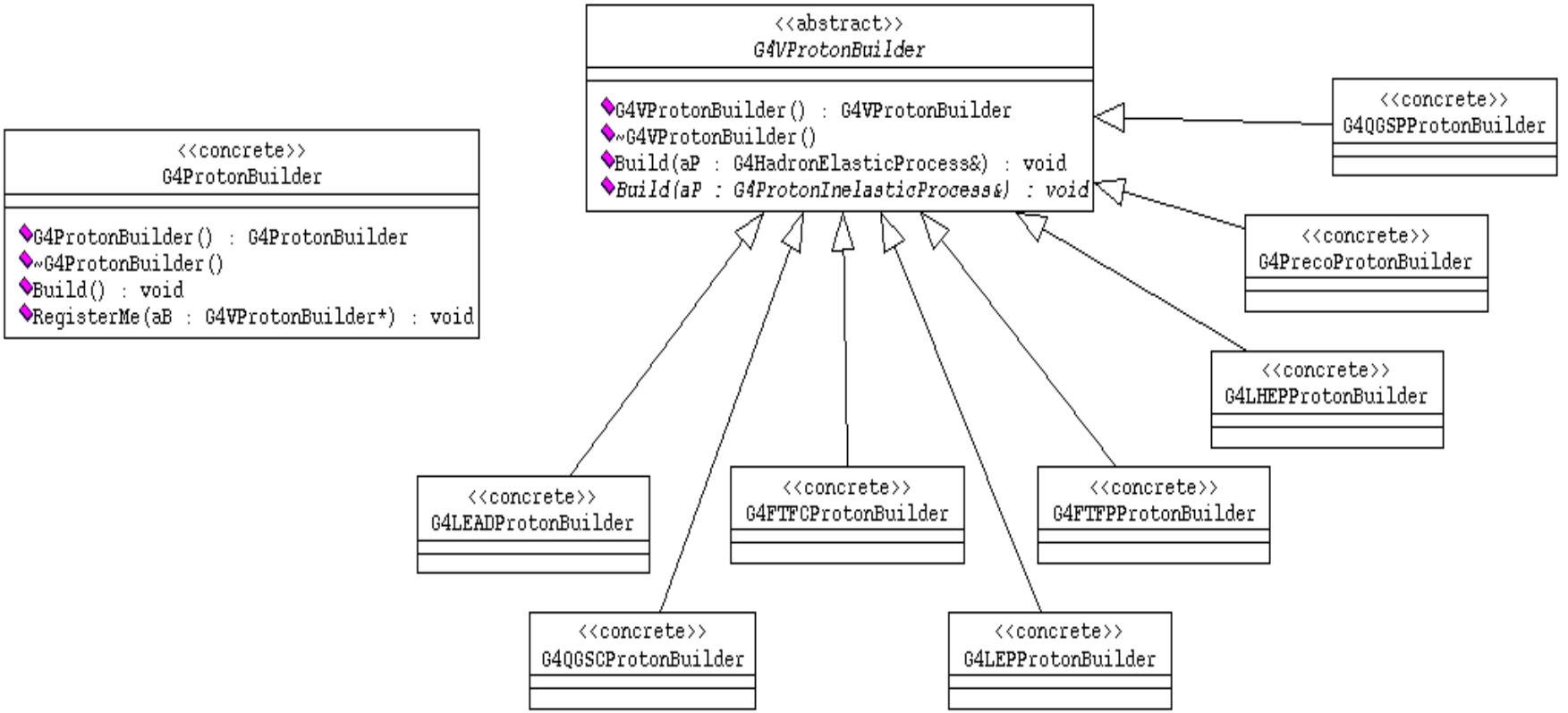
# *Design – an extension of what I had proposed during one of the last workshops, and Hisaya had realized.*



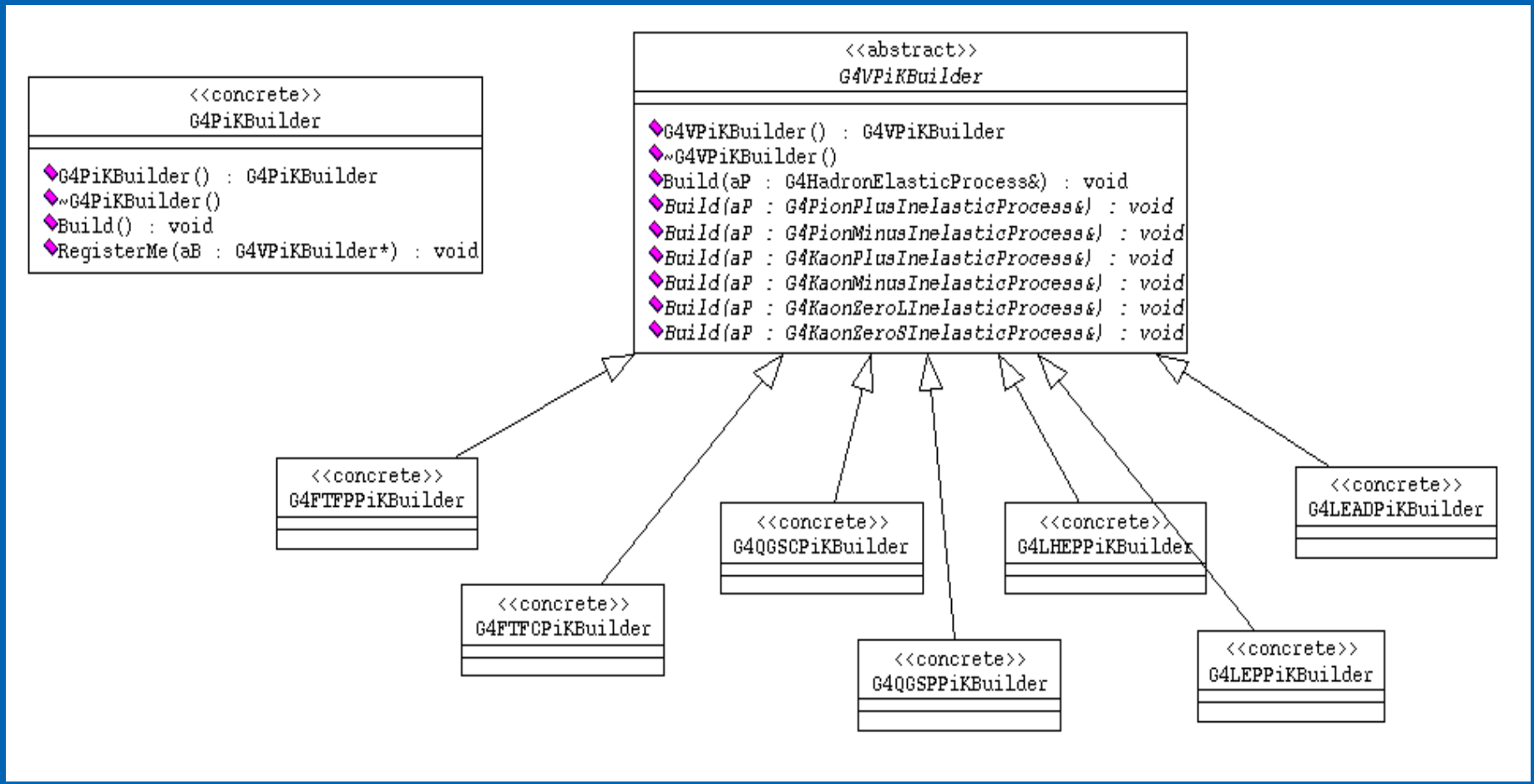
# Design

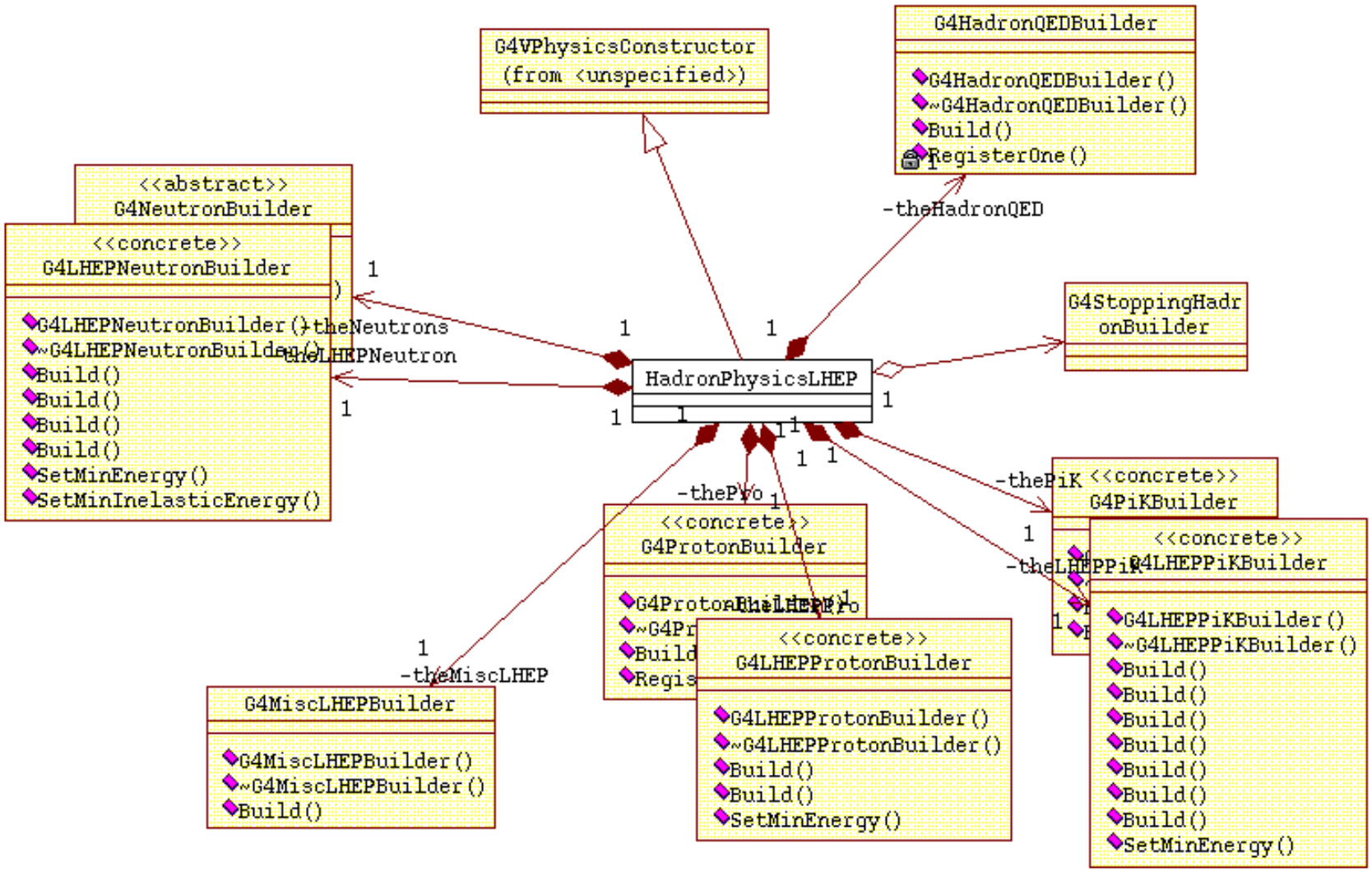

<<abstract>>
G4NeutronBuilder
G4NeutronBuilder() : G4NeutronBuilder
~G4NeutronBuilder()
Build() : void
RegisterMe(aB : G4VNeutronBuilder*) : void
<<abstract>>
G4VNeutronBuilder
G4VNeutronBuilder() : G4VNeutronBuilder
~G4VNeutronBuilder()
Build(aP : G4HadronElasticProcess&) : void
Build(aP : G4HadronFissionProcess&) : void
Build(aP : G4HadronCaptureProcess&) : void
Build(aP : G4NeutronInelasticProcess&) : void
<<concrete>>
G4QGSPNeutronBuilder
<<concrete>>
G4PrecoNeutronBuilder
<<concrete>>
G4LEPNeutronBuilder
<<concrete>>
G4NeutronHPBuilder
<<concrete>>
G4QGSCNeutronBuilder
<<concrete>>
G4LHEPNeutronBuilder
<<concrete>>
G4LEADNeutronBuilder
<<concrete>>
G4FTFPNeutronBuilder

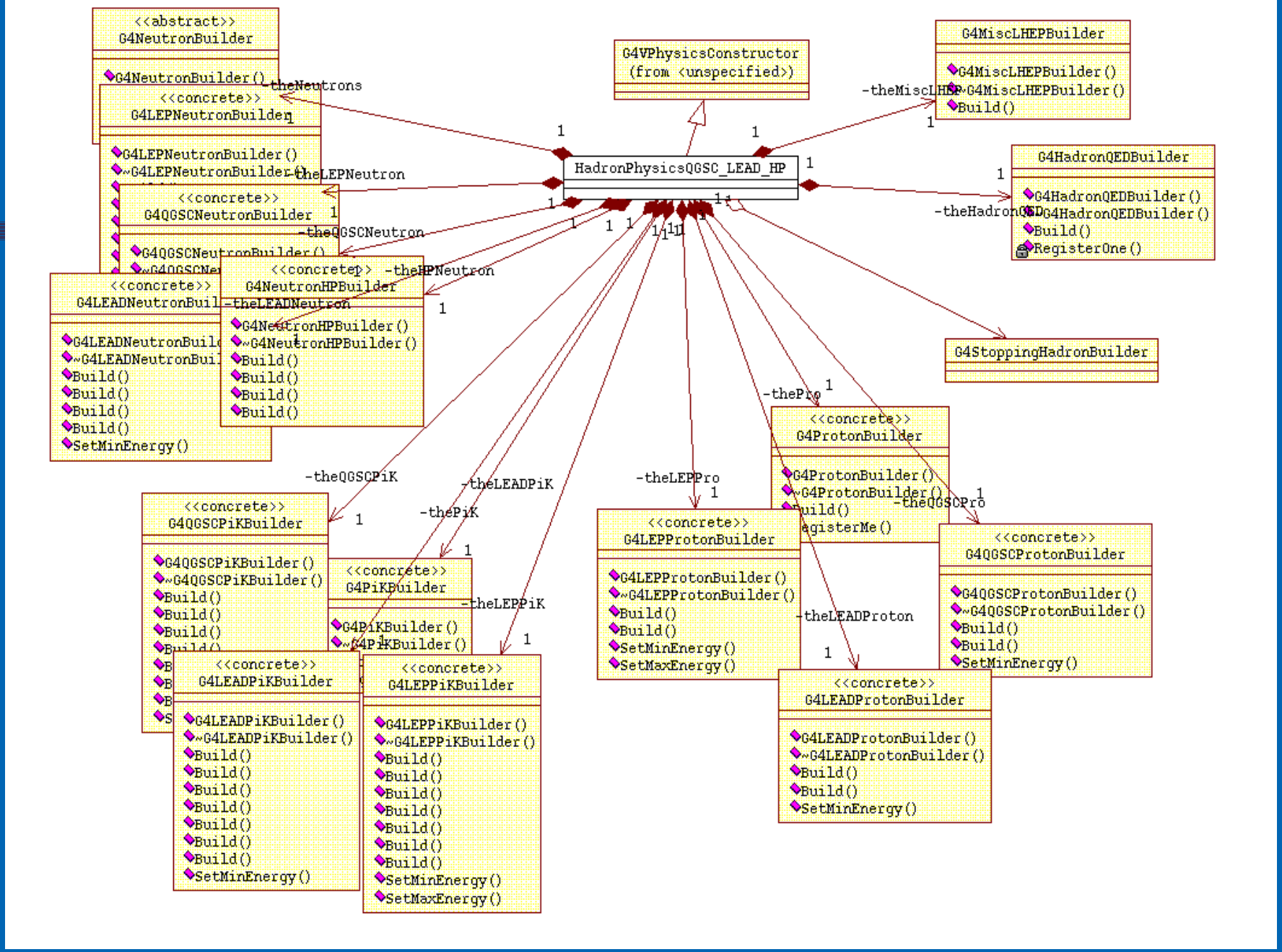
# Design


<<abstract>>
G4VProtonBuilder
G4VProtonBuilder() : G4VProtonBuilder
~G4VProtonBuilder()
Build(aP : G4HadronElasticProcess&) : void
Build(aP : G4ProtonInelasticProcess&) : void
<<concrete>>
G4ProtonBuilder
G4ProtonBuilder() : G4ProtonBuilder
~G4ProtonBuilder()
Build() : void
RegisterMe(aB : G4VProtonBuilder*) : void
<<concrete>>
G4QGSPProtonBuilder
<<concrete>>
G4PrecoProtonBuilder
<<concrete>>
G4LHEPProtonBuilder
<<concrete>>
G4LEADProtonBuilder
<<concrete>>
G4FTFCProtonBuilder
<<concrete>>
G4FTFPProtonBuilder
<<concrete>>
G4QGSCProtonBuilder
<<concrete>>
G4LEPProtonBuilder

# Design


<<concrete>>
G4PiKBuilder
G4PiKBuilder() : G4PiKBuilder
~G4PiKBuilder()
Build() : void
RegisterMe(aB : G4VPiKBuilder*) : void
<<abstract>>
G4VPiKBuilder
G4VPiKBuilder() : G4VPiKBuilder
~G4VPiKBuilder()
Build(aP : G4HadronElasticProcess&) : void
Build(aP : G4PionPlusInelasticProcess&) : void
Build(aP : G4PionMinusInelasticProcess&) : void
Build(aP : G4KaonPlusInelasticProcess&) : void
Build(aP : G4KaonMinusInelasticProcess&) : void
Build(aP : G4KaonZeroLInelasticProcess&) : void
Build(aP : G4KaonZeroSInelasticProcess&) : void
<<concrete>>
G4FTFPPiKBuilder
<<concrete>>
G4FTFCPiKBuilder
<<concrete>>
G4QGSCPiKBuilder
<<concrete>>
G4QGSPPiKBuilder
<<concrete>>
G4LHEPPiKBuilder
<<concrete>>
G4LEPPiKBuilder
<<concrete>>
G4LEADPiKBuilder


G4VPhysicsConstructor
(from <unspecified>)
G4HadronQEDBuilder
G4HadronQEDBuilder()
~G4HadronQEDBuilder()
Build()
RegisterOne()
-theHadronQED
<<abstract>>
G4NeutronBuilder
<<concrete>>
G4LHEPNeutronBuilder
G4LHEPNeutronBuilder()
~G4LHEPNeutronBuilder()
Build()
Build()
Build()
Build()
SetMinEnergy()
SetMinInelasticEnergy()
-theNeutrons
-theLHEPNeutron
HadronPhysicsLHEP
G4StoppingHadronBuilder
-thePro
<<concrete>>
G4ProtonBuilder
-theMiscLHEP
G4MiscLHEPBuilder
G4MiscLHEPBuilder()
~G4MiscLHEPBuilder()
Build()
<<concrete>>
G4LHEPProtonBuilder
G4LHEPProtonBuilder()
~G4LHEPProtonBuilder()
Build()
Build()
SetMinEnergy()
-thePiK
<<concrete>>
G4PiKBuilder
<<concrete>>
G4LHEPPiKBuilder
G4LHEPPiKBuilder()
~G4LHEPPiKBuilder()
Build()
Build()
Build()
Build()
Build()
Build()
Build()
SetMinEnergy()




J.P. Wellisch,
CERN/EP/SFT

# *Now, what does this mean ?*

- Complex problems need structured solutions !
- There are 5 levels of implementation framework.
- In the builders, we have collapsed these to level 1 and/or level 2 complexity.

- Any user can now
  - Just pick a physics list from the '*menu*'.
  - Aggregate his own cocktail from limited complexity of the builders
  - Use all 5 framework levels with their full power.

➔ A structured reduction in the level of complexity exposed to our users.

# *The WWW pages – a small demo.*

- We go to:
  - http://cmsdoc.cern.h/~hpw/GHAD/HomePage

# *Areas of usage considered*

- Goto the physics lists page.

# *Code samples*

- Go to the physics list page.

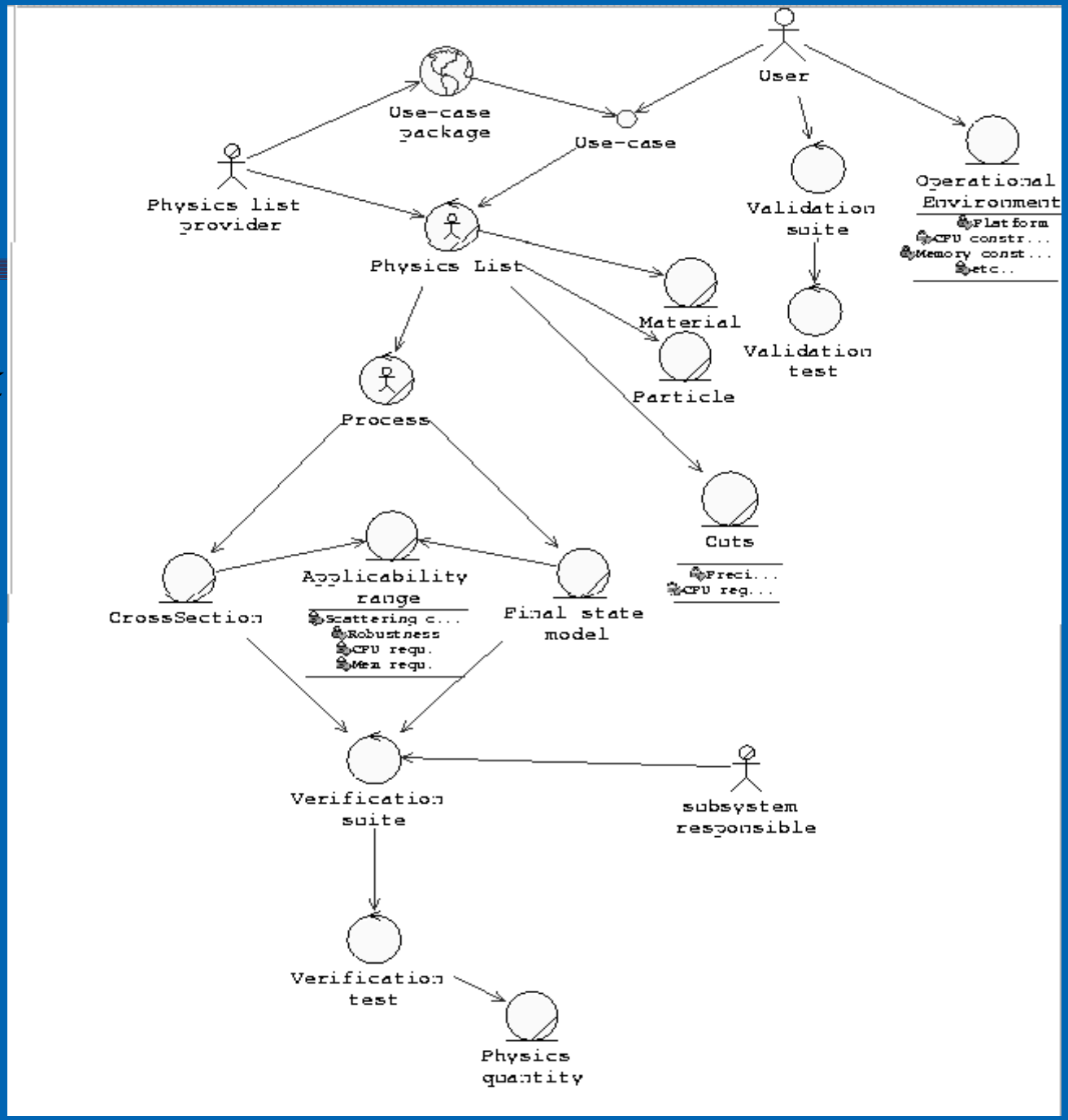
# *A non trivial effort*

- We support 14 use-case packages
- We support 16 physics lists
- They comprise 74 classes
- Their implementation needed 8740 lines of code (comments *discounted*)

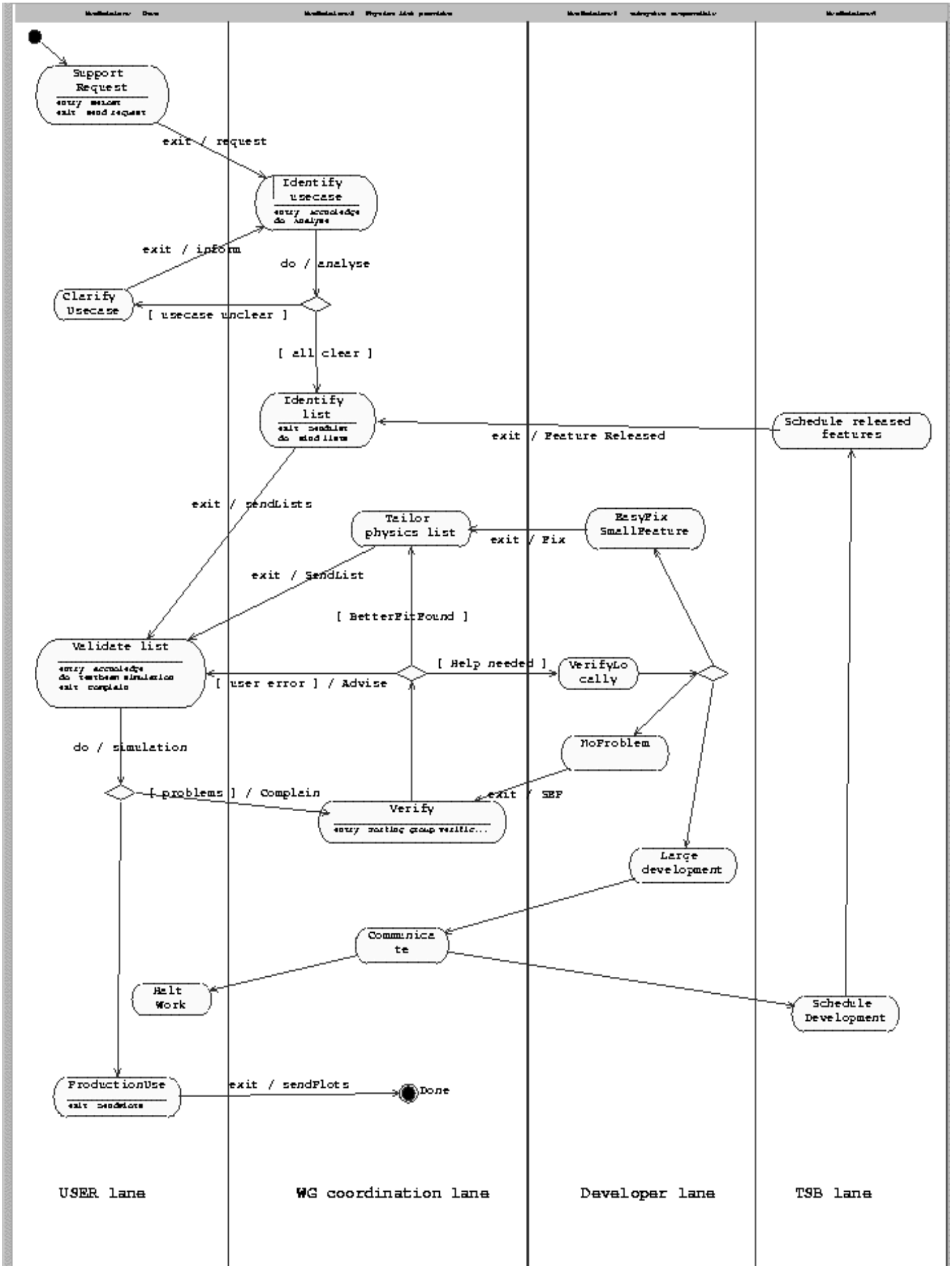
# User feed-back

- Verbatim cut and paste from some of the mails…
  - “So, to make a long story short: very USEFULL. As well, I like the idea of distributing and orgenizing the information through web pages.”
  - “There are just a few obvious mistapes, especially your name ‘J.P. Wellisch’.
  - “Thank you for this site. It is really useful.”

# The support process – static view




J.P. Wellisch,
CERN/EP/SFT


Support Request
exit / request
Identify usecase
do / analyse
exit / inform
Clarify Usecase
[ usecase unclear ]
[ all clear ]
Identify list
exit / Feature Released
Schedule released features
exit / sendLists
Tailor physics list
EasyFix SmallFeature
exit / Fix
exit / SendList
[ BetterFitFound ]
Validate list
[ Help needed ]
[ user error ] / Advise
VerifyLocally
do / simulation
NoProblem
[ problems ] / Complain
exit / SEF
Verify
Large development
Communicate
Halt Work
Schedule Development
ProductionUse
exit / sendPlots
Done
USER lane
WG coordination lane
Developer lane
TSB lane

# *Outlook*

- Much, very constructive discussions during the workshop.