Implementing models in generators

GENIE group prospective and experience

Marco Roda
mroda@liverpool.ac.uk

ECT* Trento
June 2019

Overview

- GENIE has improved its manpower
- Still we rely heavily on the community
  - Theorists – new and improved model
  - Experimenters – validation

- Adding new models is still complex as there are a number of requirements
  - Code complexity – reflex of the physics complexity
  - Adding a model is not equivalent to deploy the model
    - Nor make it usable

- Caveat
  - I will give a prospective close to GENIE point of view for obvious reasons
  - I think most of the points are general anyway

Why me to give this talk?

- I am involved in most (all?!) of the GENIE incubator projects
  - Not just in “official” reviews
  - Plenty of private (daily) conversations and support to non-expert developers
- There is not a single project within GENIE I’m not involved in
  - From cross sections to tuning
  - From Interface developments to standard maintenance
- I’m hopefully the right person to explain the issues that arises to a wider audience
  - At least to give one point of view
  - I think these should be interesting also for theoreticians – please bear with me

The right mix of skills

- Generator development requires lots of different skills
  - Theoretical understanding
  - Technical skills
    - Software engineering → Effective code structure - We are not writing macros!
    - Numerical and scientific computing skills
      - Random number computations → we are doing things by brute force or with “clever” tricks: it won’t last for long
  - Knowledge of datasets
    - This matter mostly for tuning, but you cannot put a model into a generator without a minimum amount of tuning
  - All combined with a general physics understanding

- No one in this room has expertise in all these fields
  - Different expertises are required at different stages of the implementation
    - all of them are required non the less
  - We have to accept that adding a model requires cooperation
    - Not a one man job
    - Lots of time, discussions
    - regular dedicated meetings

Reviews

- They key moment we have identified is the “Review”
  - Progress and way forward are discussed
  - executive decisions are made
  - The “no” as an answer can be regular! From all sides: physics, technical, etc
    - It’s frustrating – I know
    - But that is why we engage with people with different expertise
- Review is not code review, it’s project review
  - The typical textbook of software engineering will divide the development in
    - Analysis - 30 %
    - Design - 35 %
    - Production – actual code writing - 15%
    - Testing - 20%
- Reviews become effective only when regular
  - It allows cross checks at baby steps

GENIE development model

- GENIE has development model which is similar but physicist oriented - 4 phases:
  1) Launch – corresponds to the analysis phase
  2) R&D – combination of design and production – bulk of the
  3) Graduation – the result of the testing phase, ends up in the merge of the master branch
  4) Deployment – official release after a tuning phase

- It is geared towards fast public deployment, and clarifies the criteria that must be met (phase 1)
  - The exact scope and deliverables are agreed in advance
  - A detailed programme of work, with specific milestone used to track progress

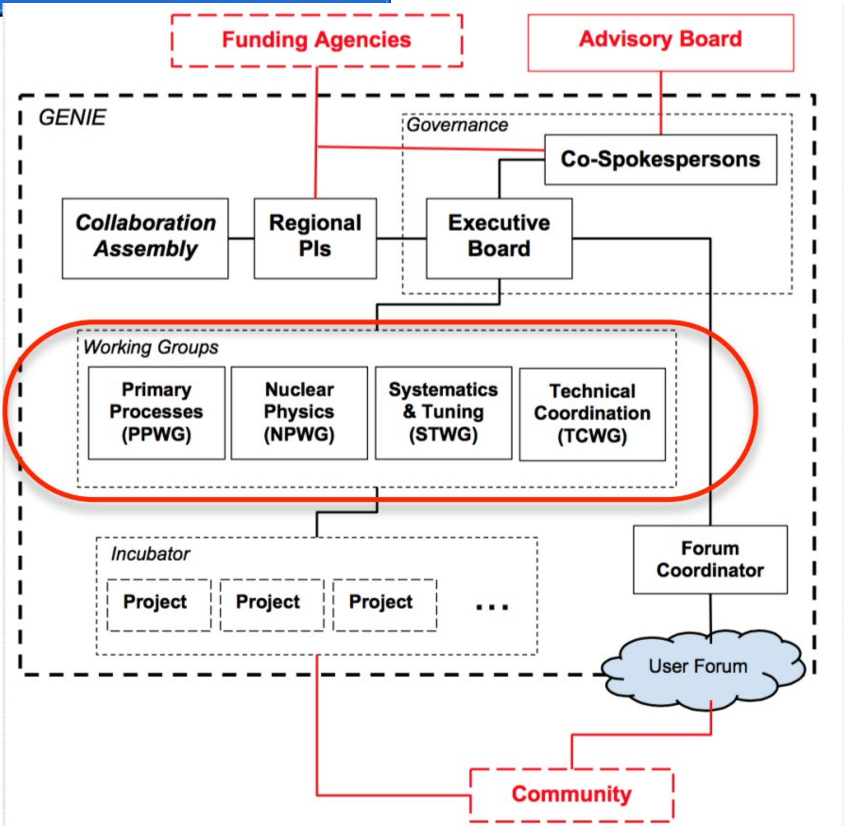
- Promotes and allows the development of sustainable software
  - The code lifetime of GENIE is ~ 20 years, and counting

- Solutions to problems are designed, not ad hoc.
  - Physics consistency
  - compatibility with the components of the infrastructure

- Well-specified review points allow timely expert feedback.The specific physics case has requirements

GENIE Incubator status

- Dedicated page https://hep.ph.liv.ac.uk/~costasa/genie/incubator.html
- A bit of statistics
  - 22 projects active now
  - 35 successfully finished
- Each incubator is reporting to one of 4 working groups
  - Each with a different chair
    - Hugh Gallagher
    - Steve Dytman
    - Costas Andreopoulos
    - Robert Hatcher



Can we do better? If yes, how?

- Factorized model
  - For an unknown reason to me factorizations is now the fashion
    - It’s not clear at which level we want more factorization (packages, apps, API?)
  - Factorization of physics is unlikely at best
    - The historical separation of nuclear model, primary interaction, hadronization and FSI is already violated in a number of model implementation
    - In fact we tent to go toward more complete models
  - This is not going to make the implementation of model easier in any way
    - The complexity of the implementations comes from the entanglement of the physics
    - Hard to “factorized” - See-saw mechanism between factorization and physics consistency
  - The physics used to be factorized in different software 20 years ago
    - generators were only simulating few processes and they were completely factorized – it didn’t last

Can we do better? If yes, how?

- Engagement, engagement, engagement
  - My personal experience suggests that regular weekly or biweekly reviews are the way
  - Seems sustainable in terms of time
  - As part of a well defined group / incubator project
    - Whatever else is equivalent to be alone

- People
  - The professional we lack most are people on the software engineering side
  - Are we interested – as a community – to train students for 6 – 12 months
    - only on coding?
  - They need to be physicists, we don't need just computer scientists
  - Long term goal

- Documentation
  - The GENIE manual is already quite long, yet not complete
  - Extensive documentation is required whenever the new models are added
  - This can speed up the integration of a model if the "surrounding" models are well understood

Lack of engagement

- The biggest issue I have noticed is the lack of engagement
  - A number of projects starts then loose momentum
    - It is my naive guess that this happens because the project seems to big and scary
  - Or people work on their own and the result is a sub-standard development
    - That cannot be adapted and most likely it has to be done from scratch

- There is not much we can do if communication stops

- Technicalities of a big project can be annoying at best
  - That’s the service part of our job
  - That is the goals are discussed at the beginning: it’s not a trial
    - We need to identify what can be asked to a developer
    - The steps are tailored on each one expertise
  - Reviews in which no progress has been made are perfectly fine

Outline

- We have increased engagement from theorists
- The development model is not likely to change
  - It is working very well when respected
- Many new features close to release
  - Hadron tensor factorization
  - SusaV2
  - INCL++
  - QuasiElastic using spectral functions