Geant4 Bindings to Python/Julia/etc

Ben Morgan

WARWICK
THE UNIVERSITY OF WARWICK

What do we mean by a binding?

- Basically, a way to call (here) a C++ interface in a library from another language
  - *Today we focus on Python and Julia, but in principle there are more*
- Why might such a thing be implemented?
  - *(Possibly) easier programming (no build/link cycle), use via a REPL*
  - *Broader experience/knowledge of binding language*
  - *Interoperability with (possibly) wider range of tools/systems (GUI, analysis, web/online)*
- **Today, want to offer some viewpoints on if/how these might be implemented for Geant4, and how we, the Collaboration, might support community interest and efforts towards this**
  - *Gathering use cases and requirements*
  - *Building a community of interested users/developers, and "incubator" projects*
  - *Long term support and sustainability*

Why not have a binding to language-du-jour?

- **“C++ is too hard”** is just moving the problem of user knowledge elsewhere
  - *Though it’s entirely on Geant4 to make our C++ interface and I/O as clear and user friendly as possible, evolving it if/as needed*
  - *… and using the valuable feedback on this that bindings may provide*
- **“All the cool kids have one/this language will save the world”**
  - *Better to ask: what use cases/requirements make a binding necessary?*
  - *… and are these well motivated and towards common goals*
- **Expectation management**
  - *Balancing competing/divergent requirements (does one size fit all?)*
  - *Performance/Validity vs usability (full API vs partial binding, PGIL)*
  - *Support, maintenance and sustainability over long term*
- **Clear that requirements and use cases are key, and need to be enumerated from all interested parties before anything else!**

If Geant4.{py,jl,...} is needed, how to realize them?

- Gathering **use cases** and **requirements** through user community is **essential**
  - *What is **actually** needed/wanted vs what is technically feasible*
  - *Identify commonalities and **divergences***
  - *Determine level of interest, **number of people willing and able to contribute to development***
- **Last point is key: at present, Geant4 Collaboration does not have FTE available to develop/support binding(s)**
  - *We **are** aware that there is interest in bindings and so we **do** want to support these and efforts towards implementing them*
- ***Today, want to outline ideas for building a community of people interested in using Geant4 from Python/Julia/etc, and development towards common solutions - to gain your feedback as a first step***

Step 1: Gathering Interest and Requirements

- *NB: I’m from a HEP background, so I’m going on what I’m familiar with - should **not** be taken as excluding other communities like Medical, Space.*
- **Use model of “Interest Group” from the HEP Software Foundation**
  - *Hold “Kick off” ½-1 day **virtual** workshop(s) for people to meet, present current implementations or ideas, discuss problems, solutions?*
  - *Wide advertisement and care in timing would be critical to ensure inclusion of all user communities and timezones!*
  - ***Deliverables**: **List of requirements, existing solutions, counting level of interest/FTE for development***
- Geant4 Collaboration would act as a concierge for organize/host these meetings, provide suitable communication channels, **but content and next steps would be led by participants in Interest Groups**
  - *NO requirement for Geant4 membership or contributor status!*

Step 2: Determining common solution potential

- Gathering of requirements and existing implementations should allow Interest Group to iterate on and determine:
    - *If a common solution, i.e. single implementation of a Python/Julia binding would meet known requirements*
    - *If divergences make this impossible, enumerate the reasons for this, sharing knowledge*
    - *List of implementations, missing required features*
- **There would be no requirement to converge on common solution!**
    - *This is to be determined and decided by the members of the Interest Group and developers of "best" existing implementations*
    - *Would expect the IG to provide recommended implementations for given set of use cases*
- ***Geant4 Collaboration would purely act as a consultant in this process - would not be there to a force particular solution.***
    - *Though we would advise on "best" use/practice of the C++ interface, and how/what to test for physics performance and validation*

Step 3: Development and Sustainability

- **Our recommendation would be to develop bindings as open projects on GitHub against latest public releases of Geant4**
    - *As geant4_pybind, Geant4.jl provide excellent models for*
    - *Only public releases to ensure stability and validity of physics and performance of the underlying C++ libraries*
- Would be on project members to build a sustainable support and user community
    - *Geant4 Collaboration would support as "Incubator"-like projects*
    - ***However, we would not be a backstop** should development/maintenance stop or developers leave.*
    - *Would welcome and support developers applying for grants to sustain the projects.*
- **Geant4 Collaboration would very much welcome regular feedback to help us improve the C++ interface as well as backporting "helper"-like interfaces.**

Benefits for Geant4 and the User Community

- To bring some **optimism** to the discussion, whether or not bindings gain success/traction, discussion and work will benefit Geant4's core C++ libraries:
  - *Engage with broad field of experienced developers…*
  - *… with good feedback that may help improve the C++ interface*
  - *Maybe identify areas users struggle with, or unclear interfaces/structures, assisting documentation/training/interface improvement*
- Worst case, bindings are **not** successful/performant, but gain a clear picture of ***why***
  - *Clear answer when asked why Python/Julia/etc not provided or don't work well.*
  - *Can enumerate how to "call" Geant4 from a given language if absolutely required, or with workarounds (e.g. how to bind/run a Geant4* ***application*** *in Python/Julia)*
- Organizing as Interest Group and (maybe) Incubator projects allows more flexibility **for both parties**, as Geant4's major stakeholders require high degree of stability, validity and support
  - *More freedom for IG to explore options, with no hard LTS/stability requirement*
  - *More freedom for different solutions, e.g. domain specific, should that be the outcome of the IG's discussion*

Questions, Discussion

- **Providing Python/Julia/etc bindings is not something the Geant4 Collaboration has resources provide itself at present**
    - *Use cases, requirements, and expertise need input from the community of users wanting these, along with expectation management of usability vs performance*
- Our proposal is to start an Interest Group between Geant4 and our users, via a first Kick-Off Meeting
    - *Led by those interested in using Geant4 from Python/Julia, with Collaboration as consultant.*
    - *To share knowledge, capture requirements and start building a core of interested and motivated developers*
- **Your comments, questions, feedback, and yes, criticism(!), are very welcome!**