

Geant-val

Dmitri Konstantinov

07.10.2024

Outline of the Talk

- **Introduction to Geant-val:**

- ▶ A brief overview of what Geant-val is and its key components.

- **Challenges and Solutions:**

- ▶ Overview of the problems in various parts of Geant-val and proposed solutions.

- **Conclusion:**

- ▶ Key takeaway

Introduction to Geant-val

- **Geant-val** is a framework designed to perform validation and regression testing for Geant4 simulations.
- It consists of three main components:
 - ▶ **Validation Suite**: A set of tests for physics validation and regression testing to ensure the reliability of Geant4.
 - ▶ **geant-config-generator**: A Python framework that automates the generation of configuration files, job submission, monitoring, and result analysis.
 - ▶ **Web Application**: A platform for uploading, storing, and comparing validation results with associated metadata.
- The goal of Geant-val is to provide a consistent and efficient way to validate Geant4 releases and track changes in simulation results.

Geant4 Validation Applications: Repository

Current Situation:

- Physics validation applications are stored in a **monolithic repository**.
- Repository URL: gitlab.cern.ch/GeantValidation/geant-validation-tests

Main Problem:

- **Desynchronization** between the Geant-val applications and original applications maintained by the authors.
- As Geant4 and validation application evolve, updates from the original authors often don't propagate, leading to outdated or inconsistent validation tests.

Possible Solutions

(A) Synchronization Process:

- **Manual or automatic synchronization** to keep the monolithic repository updated.
- **Challenges:**
 - ▶ Time-consuming and error-prone.
 - ▶ Delays in synchronization add complexity.

(B) Decentralized Repository Structure:

- Applications remain in **separate repositories** within the **same virtual organization** on GitHub.
- **Advantages:**
 - ▶ Maintains **authorship** and accountability.
 - ▶ **Easier updates** for authors.
 - ▶ Clear **git history** and flexibility with public/private repositories.

Favorable Solution - Virtual Organization in GitHub

Virtual Organization:

- A **Geant-val virtual organization** has been created on **GitHub**, with each application hosted in **separate repositories**.
- The organization will serve as the **primary repository** for the applications, though **forking the original repositories** is also a solution to ensure easier updates.

Next Steps:

- 1 Identify authors or maintainers for the validation applications.
- 2 Populate the GitHub organization with the validation applications.
- 3 Ensure clear documentation and at least one **review** per application.
- 4 Include links to presentations or papers for each application.

Geant4 Validation Web Application: Current Status

Current State:

- The web application is **up and running** at a temporary URL: geant-val-web.cern.ch.
- Earlier this year, it was migrated from **centos7** to **alma9**, which caused the temporary URL change.
- The usual URL geant-val.cern.ch will be re-activated soon (expected this week).

Technical Details:

- The application is still using **old versions of Node.js and Angular2**.
- I have not yet managed to perform a full migration this year.

Front-End Rewrite:

- A CERN technical student worked on re-writing the front end to **React.js**.
- While the new interface looks promising, it is **not fully functional** and is still in a **developmental state**.

Current Geant-val UI Limitations for Comparisons

Currently, all histograms and their associated metadata are stored in an SQL database. The metadata includes information such as "observable Geant4 version, physics list, initial particle, detector/materials

At present, the UI allows users to compare plots. However, users can only compare results between different Geant4 versions and physics lists.

Is this sufficient?

For many use cases, this setup works. However, while working with Loic, we realized that when users want to compare histograms where certain simulation parameters vary, this approach becomes limiting.

Workaround for Parameter Storage

- **Workaround:**

The current design forces us to embed additional parameters into the physics list. For example:

- ▶ `FTFP_BERT+PAR1=value1+PAR2=value2`
- ▶ `FTFP_BERT+PAR1=value11+PAR2=value22`

This workaround is a limitation of the SQL database structure, requiring parameters to be embedded into the physics list, which can complicate data management and limit flexibility.

Real Solution for Parameter Storage

- **Real Solution:**

To address this limitation, we may consider switching to a NoSQL database like MongoDB, where each record is stored as a document (BSON) with associated key-value pairs. This structure allows much greater flexibility in managing parameters and metadata, enabling you to store parameters independently.

In this case, different test results can have different metadata, which can be easily updated.

However, to implement this solution, the UI (frontend logic) would need to be completely rethought and recreated from scratch.

Geant-config-generator: Problems and Solutions

Problem 1:

- single-threaded mode only: no easy switch to multi-threaded running without manual configuration changes.

Solution

- implement an option to easily switch between single-threaded and multi-threaded modes without needing to rewrite configuration files.

Problem 2:

- desync between geant-val and the original applications in terms of configurations and analysis scripts

Solution

- use configurations and analysis scripts directly from the original application:
 - ▶ store configurations and analysis scripts in a dedicated `.geant-val` directory within the original validation application.
 - ▶ ensure that analysis scripts follow certain standards to make their usage straightforward from geant-config-generator.

Geant-config-generator: Problems and Solutions II

Problem 3:

- Job submission limited to CERN HTCondor: Currently, job submission is only supported through CERN HTCondor, which restricts usage to people with access to CERN lxplus, preventing others from using their own computing clusters.

Solution:

- extend the job submission functionality to support additional batch systems based on real use cases.
- the configurations generated by geant-config-generator are essentially a set of scripts that set all the necessary environment variables and run the simulations. These scripts can be easily run locally or interactively using Linux's parallel command, providing flexibility for users who don't require batch systems.

Geant-config-generator: Problems and Solutions II

Problem 4:

- Execution in batch on CE (alma9) only: Jobs are currently executed directly on CE (alma9) in the batch system, but running older precompiled Geant4 releases (built for centos7, alma8) requires a containerized approach.

Solution:

- Implement containerized runs using HTCondor's built-in functionality to support older precompiled Geant4 releases for centos7 and alma8, ensuring compatibility across different systems.

Need for a Dedicated Working Group

- **Current Situation:**

- ▶ If the Geant-val activity is recognized as important within the collaboration, then we need to invest more resources to achieve its full potential.
- ▶ However, at the moment, I am the only person actively contributing to the Geant-val core.

- **Challenges:**

- ▶ The workload for maintaining, improving, and further developing Geant-val is important and needed.

- **Proposal:**

- ▶ Create a dedicated working group to ensure the development, maintenance, and expansion of Geant-val.
- ▶ Alternatively, one of the existing working groups could take over the responsibility for Geant-val mechanics and its future development.

Transformation: Physics Validation TF to WG?

Proposed Structure:

- Coordinators of the group will oversee the entire Physics Validation WG.

Hierarchy:

- Representatives for each key domain:
 - ▶ High-Energy Physics (HEP)
 - ▶ Electromagnetic Physics (EM)
 - ▶ Hadronic Physics
 - ▶ Nuclear Physics
 - ▶ Medical Physics
 - ▶ DNA Community
- Geant-val Infrastructure Developers and Maintainers:
 - ▶ responsible for managing the Geant-val web application, repository, and Geant-config-generator framework.
- Application Reviewer Team:
 - ▶ Responsible for reviewing applications from both physics and performance perspectives.