Assessment of Geant4 Software Quality

What has been done so far

Elisabetta Ronchieri, Maria Grazia Pia

CERN, 7 August 2017

Background

Existing literature shows various works about the use of metrics with different goals, such as

- *improving software quality (e.g., measuring usability, efficiency and maintainability)*
- *improving reliability (e.g., measuring mean time to failire)*

Most of these studies

- are based on qualitative considerations (e.g., expert judgement)
- often lack of quantitative analysis
- do not provide metric thresholds or ranges: if they do, their values can depend on specific domain and programming language characteristics
- are applied to academic software and a sub-set of software releases

Objectives

Define a research methodology in order to

- assess the applicability of software quality metrics to large scientific software
- determine statistical methods that contribute identifying code issues and suitable metrics
- identify metrics suitable for certain software design

Work done so far

Assessment of various software quality tools (free and under licenses)

- Selected Imagix4D to measure several product metrics at different levels
- Under open evaluation license, extended three times

Identified C++ metric thresholds from literature

- **Cons: Domain specific**

Developed code to manipulate raw data of code measurements

Evaluated statistical methods to analyze data

- i.e., trend and inequality analysis

Levels	#Metrics
Variable	4
File	22
Class	24
Directory	21
Namespace	4
Function	21

Metrics at class level

Metric Name	Category
Class Hierarchy and Local Vars Coupling	Size
CK Coupling Between Objects, CK Depth of Inheritance Tree, CK Lack of Cohesion of Methods, CK Number of Children, CK Response For a Class, CK Weighted Methods per Class	**Object Orientation**
Depth of Derived Classes	Object Orientation
External Methods and Coupling	Design
Fan In of Inherited Classes	Design
Halstead Intelligent Content, Halstead Mental Effort, Halstead Program Difficulty, Halstead Program Volume	**Complexity**
McCabe Average Cyclomatic Complexity, McCabe Maximum Cyclomatic Complexity	**Complexity**
Methods Called – External, Methods Called - Internal	Design
Number of Member Attributes, Number of Member Classes, Number of Member Methods, Number of Member Private Methods	Size
Number of Member Types, Number of Total Members	Size

Chidamber & Kemerer (CK)

Metrics at file level

Metric Name	Category
Comment Ratio	Size
Halstead Intelligent Content, Halstead Mental Effort, Halstead Program Difficulty, Halstead Program Volume	**Complexity**
Inclusion – Directly Included Files, Inclusion – Files Where Directly Included, Inclusion – Files Where Transitively Included, Inclusion – Transitively Included Files, Inclusion – Transitively Included Lines	Design
McCabe Average Cyclomatic Complexity, McCabe Maximum Cyclomatic Complexity, McCabe Total Cyclomatic Complexity	**Complexity**
Maintainability Index	**Complexity**
Number of Declaration in File, Number of Functions in File, Number of Variables in File	Size
Size – Bytes, Size – Lines of Comments, Size – Lines in File, Size – Lines of Source Code, Size – Number of Statements	Size

Software candidate: Geant4

Considered several Geant4 releases from very beginning version up to 10.4.beta

wget data from the public source repository

Recalculated metrics by using a **new version of Imagix4D 8.1.4**

- Came across bugs in Imagix4D
- Received bug fixes for the open evaluated license

- **Checked raw data in a random sample**
- Met some issues in measuring metrics from 0.0 up to 3.2 due to lack of some external dependencies
  - Fixed in the last data production
- Used different metric implementations for the same metric to assess their impact on the analysis
  - Complexity, Lack of Cohesion and Maintainability Index
- Raised interest from other research groups
  - like the Atlas Software Quality group

Statistical Methods

Identified statistical methods to perform quantitative analysis

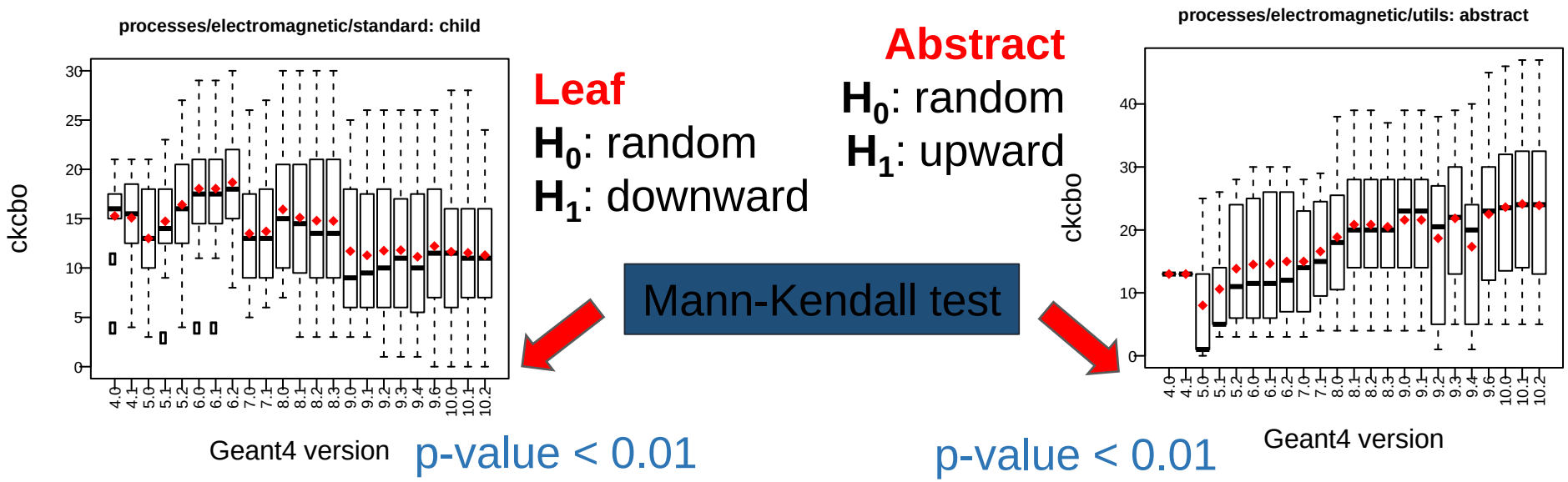
- **Trend analysis** to get information about the evolution of software to gain insight to the past and to use knowledge to shape the future
- **Inequality analysis** to get information about the aggregation of software properties

Produced over 26581 plots on Geant4 with the first data production

- considering metrics at file and class levels
- without the releases from 0.0 up to 3.2

Results from Trend Analysis

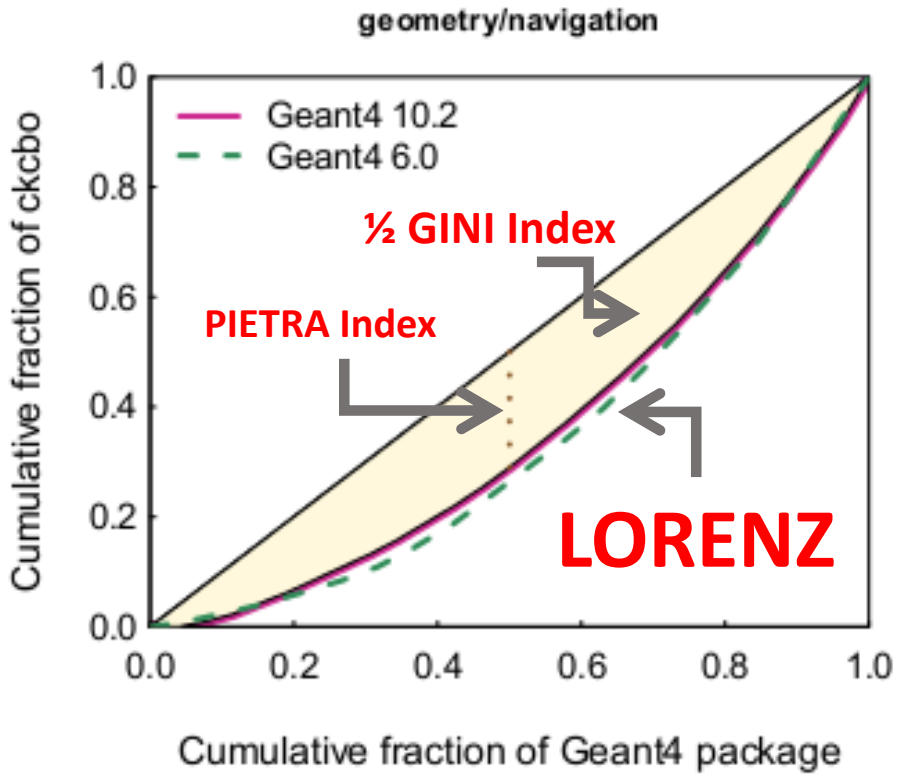
- Test for randomness non parametric
  - Mann-Kendall test
  - Cox Stuart test
  - Bartels test
- Statistical inference for upward/downward trend



From Maria Grazia, INFN Genova, presentation at IEEE NSS/MIC 2016

Results from Inequality Analysis

- Gini index
- Pietra index
- Theil index
- Atkinson index
- others



From poster at IEEE NSS/MIC 2016

In progress

- Focus on trend analysis
- Exploration of data analysis methods to aggregate trend information
- Evaluation of systematics
  - Metric implementations
  - Different trend tests
  - Metric categories
- Assessment of function metrics
- Changepoint detection: exploration of methods and feasibility study
- Correlations

Conclusion

- Performed various explorative studies on quantitative analysis of metrics
- Assessed feasibility and capability of producing valuable density profile indicating the interfaces between the various phases as well as the amount of sand results
- During this week we are going to finalize a first set of results in view of a coming conference presentation
  - Comments and suggestions are welcome *(will be duly acknowledged)*
- It would be interesting to apply the same methodology to other scientific software
  - Suggestions are welcome
  - Collaborative effort with code developers