

## Component approach to HEP Monte Carlo simulations: example of PHOTOS.

*Thursday, 16 February 2006 14:40 (20 minutes)*

Solving the ‘simulation=experiment’ equation, which is the ultimate task of every HEP experiment, becomes impossible without computer simulation techniques. HEP Monte Carlo simulations, traditionally written as FORTRAN codes, became complex computational projects: their rich physical content needs to be matched with the software organization of the experimental collaborations to make them a part of large software chains. The experimental collaboration software intrinsically use the component approach: the simulation of physical events and detector responses is typically performed in steps, by a set of dedicated software packages (such as PYTHIA, GEANT), which may be recognized as high-level “components”, interacting with each other by means of common data structure: the event record. The component approach to computer simulations is a widely discussed topic, yet not in the branch of HEP MC simulations. In this paper we try to describe the general view and approach, and the problems encountered in this area nowadays: we stress the importance of validation and testing methodology. PHOTOS Monte Carlo is used as an example of a compact package used worldwide; MC-TESTER-based method was developed for its tests and validation. PHOTOS may also serve as an example of a dialog between theoretical and experimental physicists. The involvement of the software-development experts in this dialog, in the spirit of common understanding, would certainly be beneficial to establish viable architectures for future HEP MC simulations, and still be flexible for rearrangements motivated by future (unknown at present) physics requirements.

**Primary author:** Mr GOLONKA, Piotr (INP Cracow, CERN)

**Co-author:** Prof. WAS, Zbigniew (INP Cracow)

**Presenter:** Mr GOLONKA, Piotr (INP Cracow, CERN)

**Session Classification:** Software Components and Libraries

**Track Classification:** Software Components and Libraries