Proposal for a
Standard Time Series File Format in HDF5

A. Pfeiffer[*], I. Bausch-Gall[#], M. Otter[*]
[*] Institute of System Dynamics and Control, DLR Oberpfaffenhofen,
{Andreas.Pfeiffer, Martin.Otter}@dlr.de
[#] Bausch-Gall GmbH, Munich, Germany,
Ingrid.Bausch-Gall@bausch-gall.de

Many simulation programs store their simulation results in an own file format with different information, some store only the results, other more information as unit and name of the signals. However, information supplied in these files is mostly not complete, reading the files can be inefficient and storage of huge amounts of data is almost impossible.

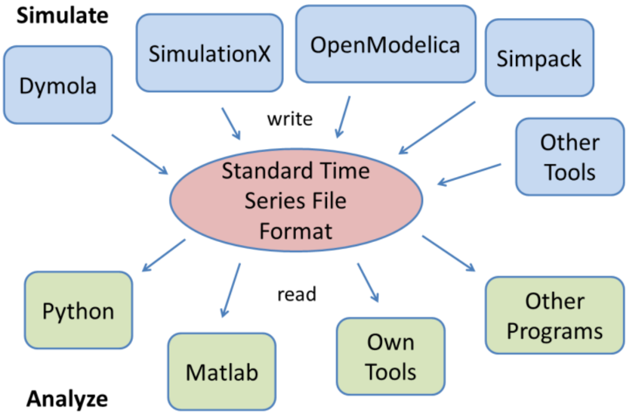

Figure 1: Standard time series file format and its interaction with tools.

These issues exist since decades for almost all simulators in many physical domains. Scripting tools such as Matlab, Scilab or Python are better suited to automate plotting of results with fine control of the layout, to generate standardized result evaluation reports, to perform signal processing (e.g. FFT), to compare with measurements, to run Monte Carlo simulations, or to perform optimization over many simulations etc.

The basic problem is then how to connect a simulation with a scripting environment. With a *standardized time series file format*, the approach from Figure 1 simplifies the task a lot, since simulation environments could generate files in this format and scripting tools may directly read files in this format.

The paper describes a proposal for a standard to store time series on file as they appear as results from dynamic model simulations (see Figure 2 for an example). We explain, why it is necessary to develop such a standard, which are the requirements to efficiently store simulation results and why we selected HDF5. The selected common basic data format, the information stored and data for time series are described in detail. At the end, we present first performance results in Python and Matlab. For example, a result file with more than 200 GBytes has been successfully generated.

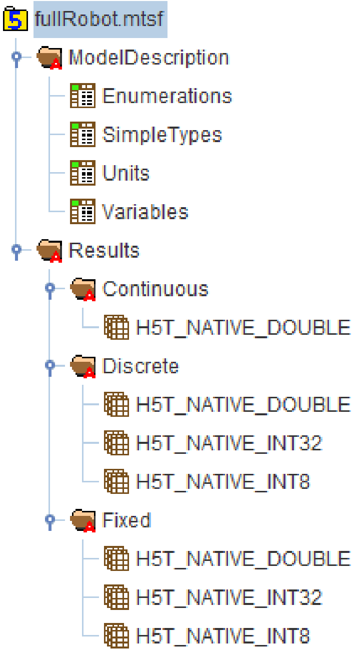

Figure 2: HDF5 hierarchy of the proposed file format.