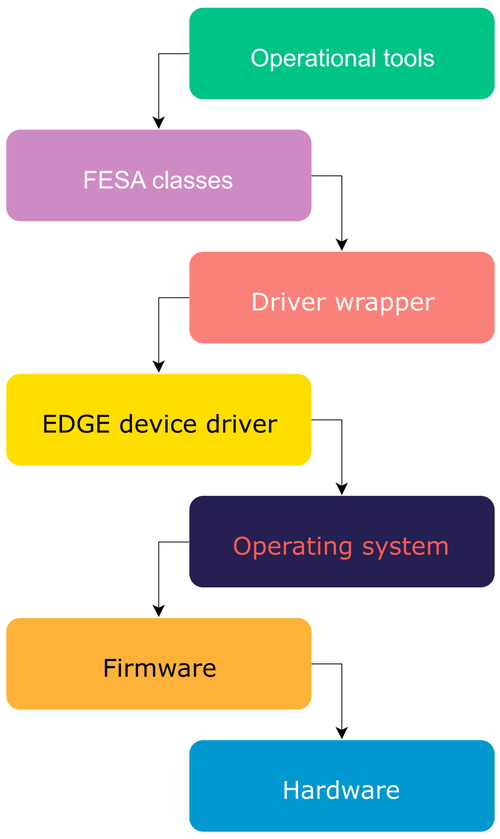
Code Generation Tools and Editor for Memory Maps

P. Plutecki, B. Bielawski, A. Butterworth
CERN, Geneva, Switzerland

Abstract

Cheburashka, a toolset created in the Radio Frequency Group at CERN, has become an essential part of our hardware and software developments. Due to changing requirements, this toolset has been recently rewritten in C++ and Python. A hardware developer, using the graphical editor, defines a memory map, which is subsequently used to ensure consistency between software and hardware. The memory map file is an input for a variety of tools used by the hardware engineers, such as VHDL code generators. In addition to aiding the firmware development, our tools generate C++ wrapper libraries. The wrapper provides a simple interface on top of a Linux device driver to read and write registers by exposing memory map nodes in a hierarchical way, performing all low-level bit manipulations and checks internally. To interact with the hardware, a software that runs on a front-end computer is needed. This toolset allows us to generate FESA (Front-End Software Architecture) classes with parts of the operational interface already present.

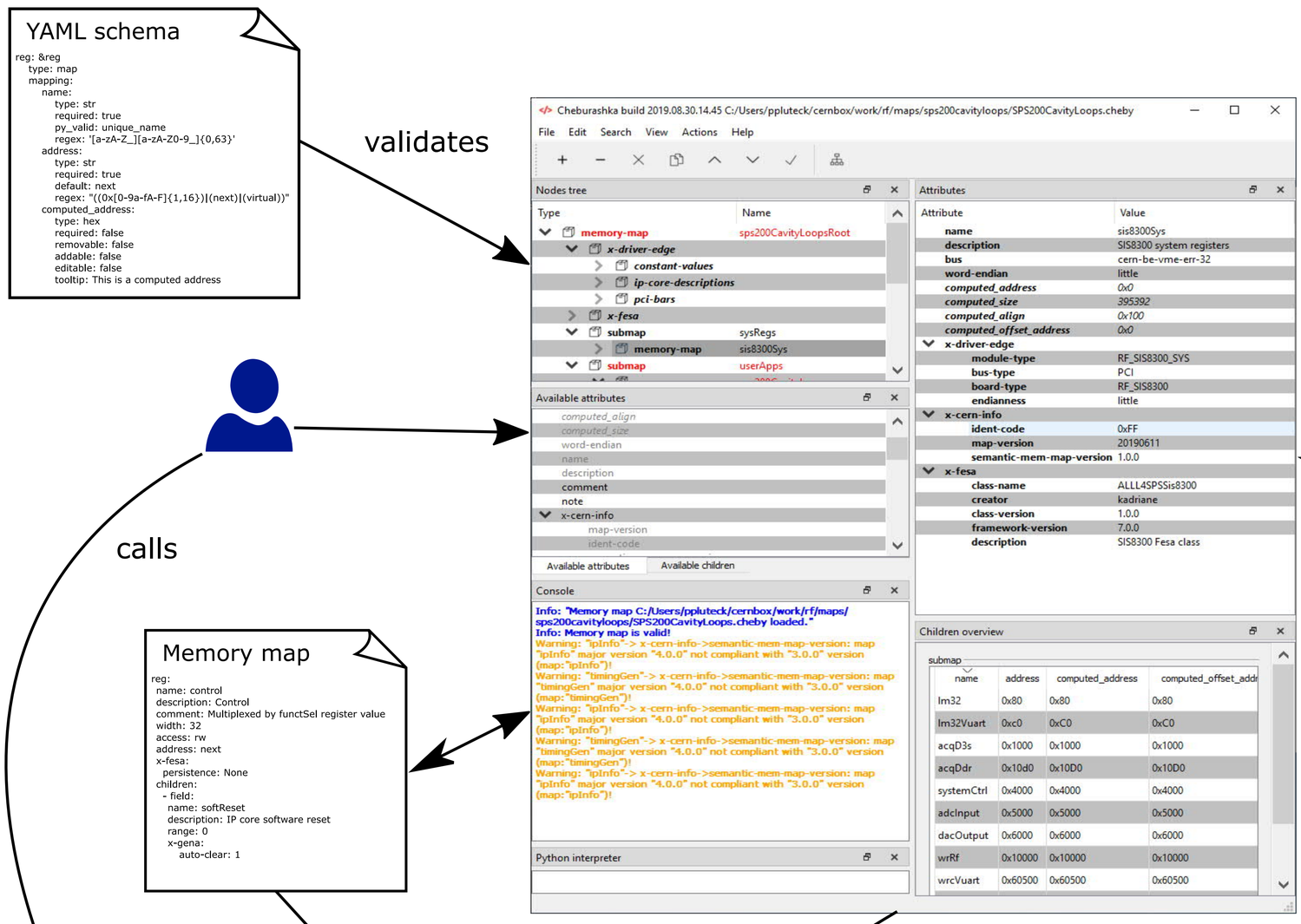


Memory Map Editor

The new GUI (Graphical User Interface) has been developed to work with new file format (YAML) for memory maps. Since many code generation tools written in Python were already production ready, instead of having a monolithic program for editing memory maps and generating code, it has been decided to split the editor into two parts:

- a generic C++ core, that, based on a schema file, can work with a single document YAML file,
- a set of Python scripts that will provide features specific for the memory map editor and possibility of running external Python tools, such as code generators.

To achieve this, a Python 3.6 interpreter has been embedded in the editor.
As it is not a trivial task to design an intuitive interface, the GUI is composed of dockable widgets, which can be freely rearranged by the user. These widgets can be hidden, detached, tabbed and resized. All changes done to the layout are saved in the user's configuration file, so they are persistent.



- memory nodes addressing
- submaps (included memory maps) handling
- advanced validation, not possible with standard schema rules

Code Generators

A tool called PyCheb is used as an entry point for our code generators. This tool is responsible for opening a memory map file and initial parsing, providing a comprehensive interface for other scripts. To make sure that any required changes and improvements in the generated code are easy to achieve, the Jinja2 templating engine has been used.

Based on a complete memory map, some parts of a FESA class, software that runs on a front-end computer, can be automatically generated.

CSV (Comma-Separated Values) file, which serves as an input for EDGE (Encore Driver GEnerator).

C++ library, which closely interacts with the EDGE Linux driver. The library provides a hierarchical interface over every memory node that is defined in a memory map. The interface allows software developers to read or write to registers and their fields, having all low-level bit-shifting and masking operations done by the wrapper. Before setting values, raw or high-level, they are automatically converted to the proper data type (depending on the register's signedness and its width), by rounding the value and clamping it to the limits of the underlying data type,

Conclusion

Opting for the extensive usage of Python has proven to be the right design choice, due to the rapid development process and the possibility to combine all existing and new tools in a single entry point, the memory map editor. As the complete toolset has been used by the software developers, new requirements and feature requests emerged. Many of those caused modification of the memory map schema, which, thanks to the highly configurable GUI, have been implemented in an efficient manner. Memory map attributes often require special validation rules, which now can be written in Python. The code generators, with the logic separated from the view using a template engine, have been extremely easy to customize. Since the software and hardware developers at CERN often work on different platforms, the editor and the code generators are easily accessible for both Linux and Windows users.

ICALEPCS 2019
17th Biennial International Conference on Accelerator and Large Experimental Physics Control Systems
Reference: MOPHA115