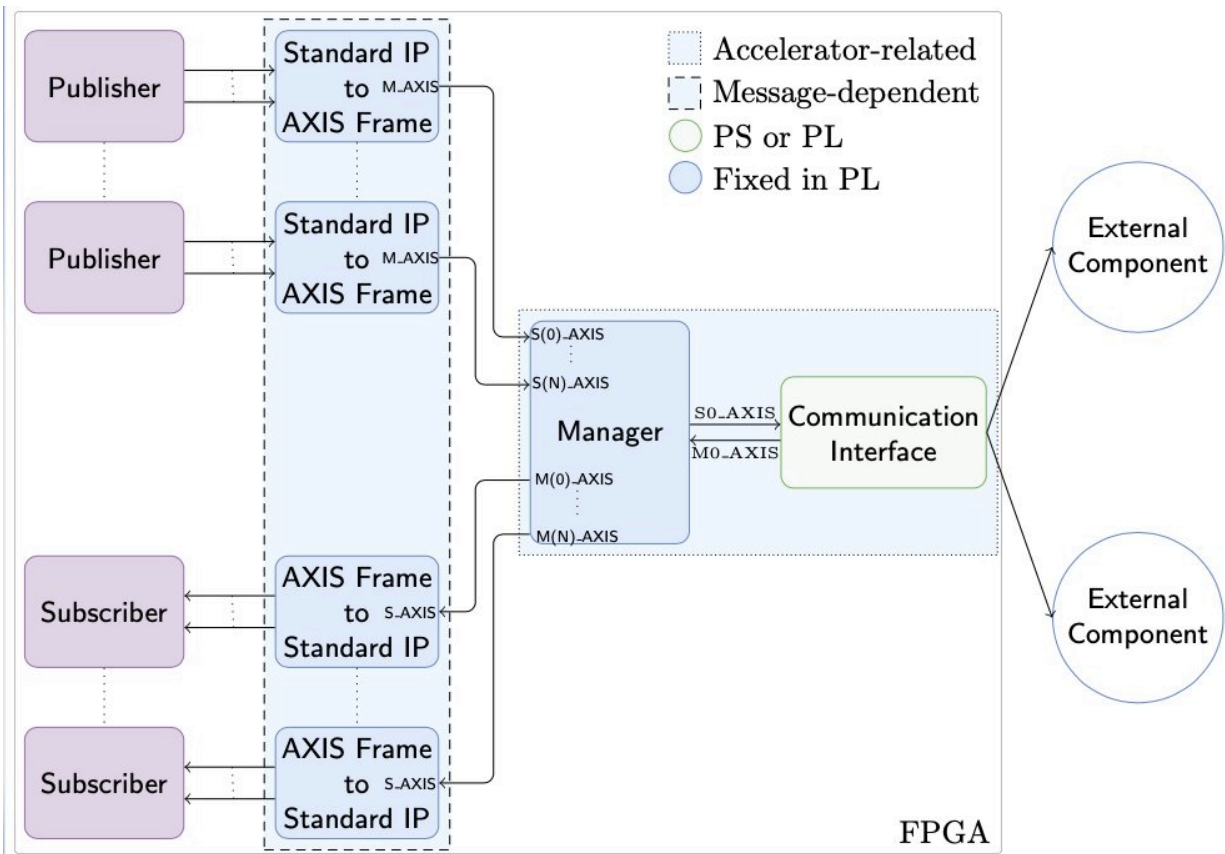


# A technique to automatically generate hardware components for robotic systems

October 21 2021, by Ingrid Fadelli



Generic hardware architecture for robotic applications of an FPGA design.  
Credit: Podlubne et al.

As robots become increasingly sophisticated and advanced, they will typically require a growing amount of hardware components, including

robotic limbs, motors, sensors and actuators. In addition, robots have integrated computers that process data collected by their sensors and plan their future actions accordingly.

Most [software solutions](#) currently running on these computers, however, are not ideal, as their speed limitations make them unable to process particularly large amounts in real time. A possible way to enhance the capabilities of computers integrated inside robots is to use field programmable gate arrays (FPGAs), [semiconductor devices](#) based around a matrix of configurable logic blocks that are connected via programmable interconnects.

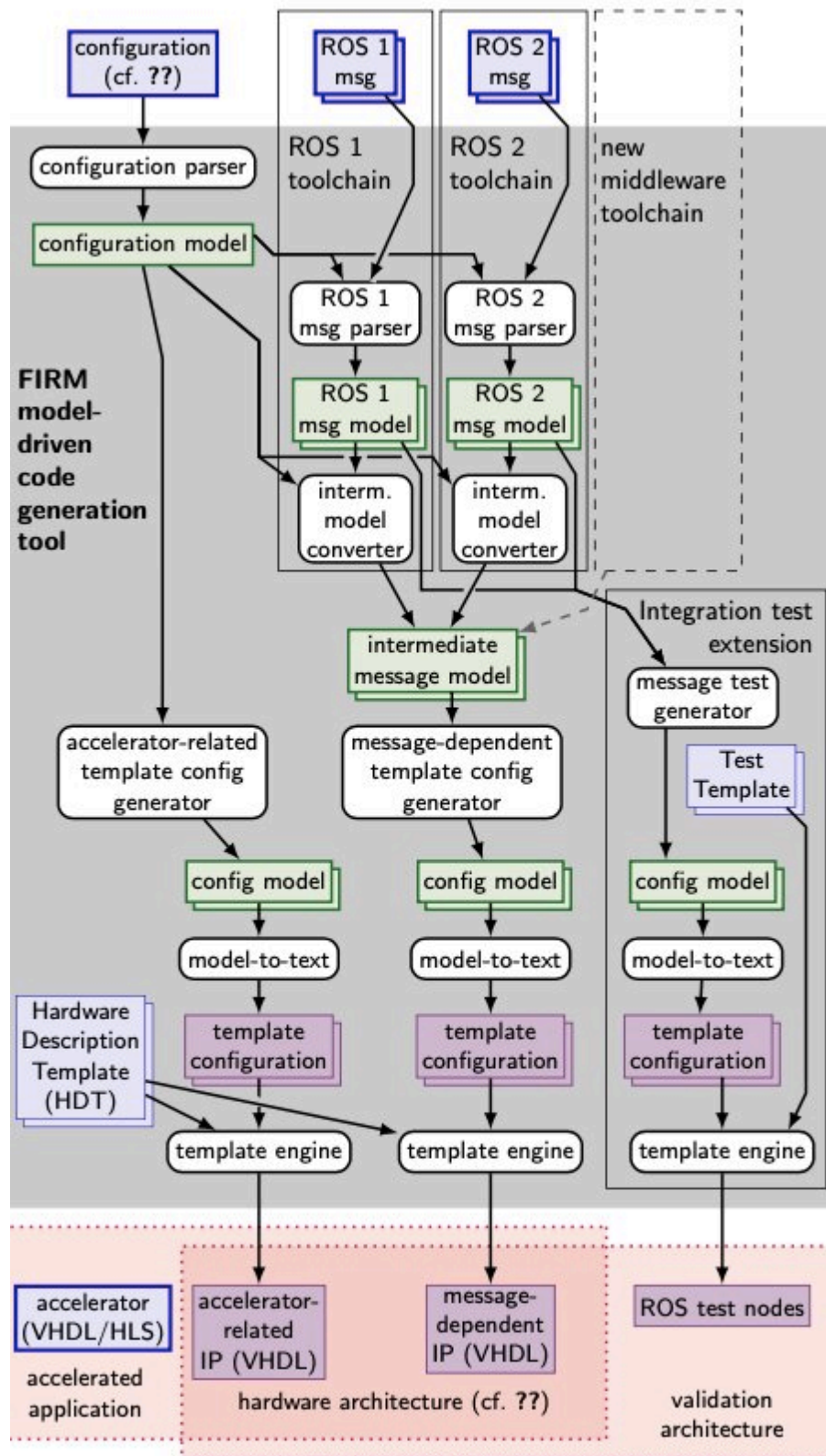
A significant advantage of these devices is that they can be re-programmed to suit specific applications. FPGAs could significantly enhance the computing capabilities of robots, while also making them more adaptable to specific applications. However, incorporating them into existing systems has so far proved to be highly challenging, as using individual accelerators with specific integration capabilities limits their applicability.

Researchers at Technische Universität Dresden (TUD) have recently developed a technique that could enable the development of robots that integrate numerous hardware accelerators. This technique, presented in a paper published in *IEEE Access*, could ultimately facilitate the replacement of existing [software components](#) powering robotic systems with components based on FPGAs.

"This work is in the context of [the CeTI project](#), which is aimed at enhancing collaborations between humans and machines or, more generally, cyber-physical systems (CPS) in real, virtual and remote environments," Ariel Podlubne, one of the researchers who carried out the study, told TechXplore. "Particularly, it is an [interdisciplinary work](#) combining embedded [hardware research](#) (Chair of Adaptive Dynamic

Systems) and [software modeling \(Chair of Software Technology\)](#)."

The new study by Podlubne and his colleagues is an extension of their previous research, which explored possible ways of integrating FPGAs into robotic systems. The approach they presented performs a thorough analysis of message specifications associated with the Robot Operating System (ROS), the ROS2 operating systems and potentially other software solutions. It then uses the results of this analysis to generate hardware interfaces and architectures for robotic systems.



A complex staged model-driven code generation toolchain is used to generate the hardware interfaces. Credit: Podlubne et al.

"Our work demonstrates the ability to generate a complex FPGA-based system from a simple description of the application, based on a known specification for roboticists (ROS messages)," Podlubne said. "With that, parts of a robotic system can be replaced by an FPGA, creating better performing and more energy-efficient systems."

The toolchain can generate all of the components necessary to create a highly performing robotic system, excluding only the accelerator logic, which will need to be programmed by developers working on the system. The new approach can thus significantly simplify the interfacing of hardware architectures and software components, which can be a cumbersome task for those creating robots.

Initially, the researchers showed that their method can generate hardware components for systems based on the ROS operating system. However, they were then able to extend its functionalities so that it also supported the ROS2 operating system.

"A complementary effort was the testing infrastructure," Podlubne said. "We went one step further to evaluate all existing ROS messages, beyond some use cases. This proved to be extremely useful as the development process involves multiple iterations to have a robust solution. Our goal was to achieve full ROS/ROS2 support, and our testing infrastructure allowed us to catch bugs and create confidence in our research."

In the future, the approach could pave the way toward the development of better-performing [robotic systems](#) based on FPGAs. These systems could be capable of analyzing larger amounts of data in real-time and might thus assist humans in solving more complex problems.

"Our next studies will focus on extending the toolchain to automate the insertion of FPGA accelerators (where the computation is performed) and include Dynamic Partial Reconfiguration (DPR) to change the accelerators on the fly, according to the current needs of the application that is deployed," Podlubne added.

**More information:** Ariel Podlubne et al, Model-Based Approach for Automatic Generation of Hardware Architectures for Robotics, *IEEE Access* (2021). [DOI: 10.1109/ACCESS.2021.3119061](https://doi.org/10.1109/ACCESS.2021.3119061)

Ariel Podlubne et al, FPGA-ROS: Methodology to Augment the Robot Operating System with FPGA Designs, *2019 International Conference on ReConfigurable Computing and FPGAs (ReConFig)* (2020). [DOI: 10.1109/ReConFig48160.2019.8994719](https://doi.org/10.1109/ReConFig48160.2019.8994719)

Johannes Mey et al, Relational reference attribute grammars: Improving continuous model validation, *Journal of Computer Languages* (2020). [DOI: 10.1016/j.col.2019.100940](https://doi.org/10.1016/j.col.2019.100940)

© 2021 Science X Network

Citation: A technique to automatically generate hardware components for robotic systems (2021, October 21) retrieved 20 March 2024 from <https://techxplore.com/news/2021-10-technique-automatically-hardware-components-robotic.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.
---