Enabling External Physical Type Annotations for Physically Relevant C++ Code Segments

(Technical Report)

A Public-Private Partnership Disaster: The Aviation Industry's Reaction to the Boeing 737 MAX Incidents

(STS Research Paper)

An Undergraduate Thesis Portfolio
Presented to the Faculty of the
School of Engineering and Applied Science
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Computer Science

by

Charlie Houghton

May 6, 2021

Preface

If errors in cyber-physical systems are found, the authority of the cyber-physical system's domain (e.g. FAA for aerospace systems) must ensure that those errors are resolved, otherwise, dependability would remain compromised. How can dependability in cyber-physical systems be strengthened?

Cyber-physical systems, such as autonomous vehicles and aerospace guidance, interface software with the physical world. Software errors can lead to expensive and catastrophic failures, so it is valuable to formally verify software correctness. Physical semantic errors occur when data representing physical phenomena are used in operations in ways that are not physically meaningful, e.g., by adding numbers that represent quantities that, in the physical world, cannot be added. How can cyber-physical systems engineers discover physical semantic errors in their programs? We developed a Visual Studio Code extension that embeds additional type information into physically relevant segments of C++ code and checks for physical semantic errors for time operations.

The 737 MAX disasters have demonstrated failures in U.S. commercial aviation safety. How, then, have U.S. airplane manufacturers, airlines, and U.S. aviation governing bodies balanced profitability and efficiency with safety following the incidents? Systemic FAA oversight failures, company-wide management deficiencies, and short-term economic incentives from airlines have pressured Boeing and other manufacturers to value cost control over safety.

List of Contents