Credible Autocoding for Verification of Autonomous Systems

Juan-Pablo Afman
Graduate Researcher
Georgia Institute of Technology
Agenda

- Introduction
- Approach
- Prototype
- Relevance
- Case Studies
- Current Work
- Conclusions
Introduction

• Formal methods of certification
  New Era → Higher standards

• There is a need for automated technology to meet these standards
  • In a timely and cost-effective manner

The solution is found at the intersection of the expert’s domain and the formal computer scientist’s
Enable the maturation and commercialization of products and procedures aimed at developing high quality safety critical embedded software
Current software development process

• Weak semantics carrying flow

  • High level properties are always known to designers

  • Data flow modeling languages do not provide a way to bring these in the software development process

  • As a result, high level properties get thrown out.
Agenda

Introduction

Approach

Prototype

Relevance

Case Studies

Current Work

Conclusions

User Requirements

High-Level Specifications

Model

Code

System Validation

Integration Tests

Unit Tests

x10000
Introduction

Approach

Prototype

Relevance

Case Studies

Current Work

Conclusions

Agenda

User Requirements

Autocoding Process

High-Level Specifications

Model

Code

Integration Tests

Unit Tests

System Validation

Verification & Validation

x10000
The code producer provides the evidences.

Code

Control Semantics:
1. Properties of Control Systems
2. Plant Models.

The certification authority checks the evidences.

Proof-Checking
1. Theorem Proving.
2. Abstract Interpreters.
3. SMT Solvers.

Proof for the Code
(Code Annotations)

Binary

Manual
Semi-Automatic
Automatic

Static Analyzer
Credible Autocoding
Certified Compiler

Agenda
Introduction
Approach
Prototype
Relevance
Case Studies
Current Work
Conclusions
Exhaustive Testing

Input space is a set of all possible controller inputs and parameter variations.

Simulation outputs: need to be verified by either human or a program.
One-shot “testing” with Credible Autocoding

Controller#1  Controller#2  Controller#3  Controller#N

Credible Autocoding

Proof Verification Backend

The input is now a “function space”
Certification Guidelines
DOA-178B/C

• DOA-178B:
  • Defines goals but not methods.
  • De Facto: run enough tests or simulations.

• DOA-178C/DO-333:
  • No longer technology agnostic
  • Mentions formal methods and model-based software engineering.
    • Abstract interpretation
    • Model checking

Figure FM6-1 Level A Software Verification Processes
(DO-178C TABLES A-3 TO A-6)
Where does Credible Autocoding Fit?

Figure FM.6-1 Level A Software Verification Processes
(Do-178C TABLES A-3 TO A-6)

Agenda | Introduction | Approach | Prototype | Relevance | Case Studies | Current Work | Conclusions

13
Prototype Technology

• Initial level of abstraction:
  • Mathematical description of system
  • State-space representation

• Assumption:
  • Designers have proof of good design at this level.
Prototype: annotated Simulink model
Prototype: Gene-Auto+

- Open source code generator (Gene-Auto)
  - Extended to support these annotative blocks

- Generates formally annotated C Code.
  - Annotations are in ACSL
  - ANSI C Specification Language
Prototype Current Status

Maturation includes addition of more high level properties:

1. Performance.
2. Robustness.
3. Nonlinear: sliding mode, sum of squares, etc.

source: quanser
Prototype Maturation

Proofs of good behavior exist at the level of mathematics. More research needs to be done for credible autocoding.

Important domain properties:

1. Convergence rate.
2. Optimality.
Matured Prototype

Advanced Controllers
(Convex Optimization,
Dynamic Programming, etc)

High-level Planner:
path planning,
machine learning,
Human-in-the-Loop

Feedback Controllers
Added value

• Expertise from both communities
  • Computer Science
  • Control Theory

• Surentez provides a credible autocoding toolchain that:
  • Allows designers to provide high level system properties at modeling level
  • Generate code with formal annotations describing these properties
  • Provides proof that design meets the system requirements
  • Eases the certification authority into accepting the validity of these annotations
Complementary!

- Lots of tools out there for formal methods
  - KeyMaera
  - Model based design / low level verification (Scade)
  - Code Coverage tools
  - Abstract interpretation, WCET (Absint)

- None to prove high level properties at the level of the code
  \[\rightarrow\] Formal Validation
Relevant to Aerospace Industry

Aircraft manufacturers currently make their safety-critical software in-house
Relevant to Aerospace Industry

Big disconnect between systems designers and computer scientists
Relevance to Aerospace Industry

• Aerospace systems cost trend is shifting away from traditional structures, aero and propulsion to software and systems.
  • Software verification is becoming one of the leading components of system cost
    • FAA flight certification
• Verification will become even larger challenge as systems become more highly integrated

source: Dr. Jim Paunicka
Boeing Research and Technology
Verification & Validation Challenge

- Advanced Controllers do not fit in the current V&V procedures
- Need innovative approaches for efficient certification of emerging technologies
  - Multi-entity systems
  - Human interactions with autonomy
  - Fused Sensor Systems
  - Mixed criticality functions dependent on information of varying confidence

US Combat Aircraft Price

Source: Norman Augustine - Aerospace Business man and Secretary of Army (75-77)
Relevance to autonomy

• Autonomous systems WILL USE these advanced controllers
  • Lots of new/unconventional control strategies...
  • ... based on new/unconventional sensing technologies:
    • Computer vision
    • Human language processing
Case Studies
Industrial Example

The annotated code runs successfully on the FADEC-in-the-loop simulation bench.

Documented the closed-loop system local stability at all operational points, and we have demonstrated that this documentation can be checked.
Current Work

• Customer assessment
  • Understand what the true needs are across different industries
  • This will allow us to focus our efforts to provide a valuable product
<table>
<thead>
<tr>
<th>Agenda</th>
<th>Introduction</th>
<th>Approach</th>
<th>Prototype</th>
<th>Relevance</th>
<th>Case Studies</th>
<th>Current Work</th>
<th>Conclusions</th>
</tr>
</thead>
</table>

Conclusions
Conclusions

• Provide framework that allows the production of software that can be easily verified.

• Promote the development and commercialization of formal methods
  
  • Expected to lead to cost reduction
  • Ensure broader share of enjoyment of cyber physical revolution
Our Proposition

• Formal verification & validation
  • High level property semantics
    • Embedded at model level
  • Mathematical proofs
    • High level properties proved at code level
    • More rigorous than test until you drop
Name: Juan Pablo Afman
Email: jafman3@gatech.edu
Phone: (423) 227-6870