KRATES
inseneribüroo • engineering bureau

AdaCore
QGen

QGen as a qualifiable code generation backend for TASTE

IB Krates OÜ

Andres Toom

Model-Based System and Software Engineering - Future directions

8 December 2016 ESA/ESTEC

What is QGen?

A qualifiable and customizable code generator

from Simulink® and Stateflow® to SPARK and MISRA C

A formal model verifier

for runtime errors and functional properties

An open and extensible framework

to integrate heterogeneous models

KRATES
inseneriburoo • engineering bureau

QGen - Brief History

- Gene-Auto open source code generator (EUREKA project, 2006-2009).
  - 11 partners, Leader: Continental Automotive France
- Project-P (France- and EU-funded collaborative R&D project, 2011-2015). 19 partners Leader: Continental Automotive France
  - 19 partners Leader: Continental Automotive France
- Currently, developed and maintained by AdaCore and IB Krates
- Available as a commercial open-source GPL licensed product
  - Please visit http://www.adacore.com/qgen

KRATES
inseneriburoo • engineering bureau

QGen - Main Features

Support for a large subset of Simulink® / Stateflow®

- Around 120 blocks, optional checks for MISRA Guidelines for Simulink®
- A safe subset of Stateflow® supported

Code generation producing MISRA C and SPARK (formally provable subset of Ada 2012)

Integrated with compilation and testing frameworks

- Integration with GNAT Pro compiler for qualified, end-to-end tool chain
- Integration with GNATemulator and GNATcoverage for structural coverage analysis (up to MC/DC) without code instrumentation executing embedded object code

Includes a static model verifier

- Finds Run-time errors (divisions by zero, overflows, ...), logical errors (dead code)
- Verifies Functional / safety properties (Simulink® assertion blocks)

KRATES
inseneriburoo • engineering bureau

QGen - Main Features (2)

Qualifiable for different safety critical domains

- DO-178C, EN 50128, ISO-26262 TCL3
- Qualification includes validation against Simulink® simulation

QGen Model-Level Debugger

- Model-Level breakpoints, stop on event, transition etc
- Correlate model blocks with generated source and assembly code

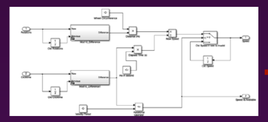
An open and extensible framework

- “The gcc for modeling languages”
- Designed to accept multiple languages in input and output, including in-house DSLs
- A single code generation style / strategy for all of your modeling languages
- XMI-based model import / export at different abstraction levels

KRATES
inseneriburoo • engineering bureau

QGen - An open and extensible framework

Traditional Code Generators

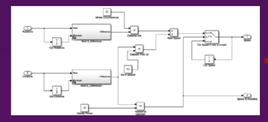


Simulink Model

Black Box

Source Code

QGen



Simulink Model

Access to intermediate representations

Modeling standard checking

Integration with UML

Additional verification

Makefile generation

Extract traceability data

Processor customization

Source Code

KRATES
inseneriburoo • engineering bureau

QGen – TASTE
for Simulink

ESA-PECS Project (IB Krates, 2013-2015)

- "Integrating the QGen Automatic Code Generator with the Space Component Model"

Improved code generation from Simulink® / Stateflow® in TASTE

- Code generation driven by a single build script, less manual steps, fully repeatable

Direct use of native data types defined in ASN.1

- Less buffers required. Cleaner glue code

Code generation with formal verification support

- CodePeer and SPARK integration

Dedicated support for on-target regression testing

- Comparison against stored simulation results. No need for external IO or software

Comparative study of the DO-178C and ECSS based qualification

- Largely compatible aims and processes. The DO-330 Tool Qualification guidelines can be used for tayloring the relevant ECSS standards

KRATES
inseneriburoo • engineering bureau

QGen – TASTE
for SDL and VDM

(1)

- **ESA-IC Project (IB Krates, 2016-2018)**
  - "Qualifiable code generation backend for TASTE"
  - Extends the previous project to new (functional) modelling languages
- **Objective 1: QGen-based universal and qualifiable code generation backend for SDL and VDM-SL**
  - Support SDL models with integrated VDM-SL guards / actions by importing the intermediate representation of the OpenGEODE (SDL) and Overture (VDM-SL) tools
  - Transform the models inside QGen reusing parts of the Stateflow® and SPARK transformation chains, as possible
  - Post-process and generate C / Ada (SPARK) code using common QGen backend facilities
- **Objective 2: Simulation and debugging of SDL + VDM-SL models**
  - Support the simulation of SDL + VDM-SL models in the OpenGEODE tool based on QGen generated code.

KRATES
inseneriburoo • engineering bureau

QGen – TASTE
for SDL and VDM

(2)

Objective 3: VDM-SL based specification and verification workflow

- Formal high-level contracts of a system given in VDM-SL
- Detailed design expressed in
  - Simulink (single component) or
  - AADL and Simulink (multiple components)
- Verification of the components can be performed in
  - Simulink (simulation conforms to contracts)
  - Generated code (static analysis of SPARK contracts)
  - Generated code (testing / run-time verification of SPARK contracts)

KRATES
inseneriburoo • engineering bureau

QGen – TASTE
for SDL and VDM

(3)

Objective 4: Support Simulink – AADL roundtrip in TASTE

- Ideally, the system's architecture is modelled in TASTE/AADL and the components are modelled in a suitable DSL, e.g. Simulink.
- However, existing complex Simulink models also model parts of the system's architecture
- Currently, it is complicated to keep the same structure on both sides
- This task is intended to refine this mapping and provide dedicated code generation support

Objective 5: Case studies

- All the developments shall be validated on case studies
  - Case studies from other ESA projects
  - 2 case studies from a Cube Satellite developed by the Tallinn University of Technology

KRATES
inseneriburoo • engineering bureau

KRATES
inseneribüroo • engineering bureau

Thank you!