The SMACCMPilot Project

A High Assurance Software Research Project

Pat Hickey

June 7, 2013

galois

Table of Contents

- What is the SMACCMPilot project
- What High Assurance means
- How we build High Assurance software
- What's flying right now

galois

What is SMACCMPilot

A Project by Galois:

- We Create Trustworthiness in Critical Systems
- Computer Science research using expertise in language design and formal methods.
- 14 years and 40 employees in Portland, Oregon.
- Named after Évariste Galois (1811-1832), genius mathemetician

galois

What is SMACCMPilot

- Secure
- Mathematically-
- Assured
- Composition of
- Control
- Models

galois

Why High Assurance?

What if we built software like we built bridges?

galois

Why High Assurance?

- Ariane 5 Flight 501 Failure
  - Failed converting 64 bit float to 16 bit integer
- Intel P5 Pentium FDIV bug
  - 1 in 9 billion computations were wrong
  - $475 Million to fix

galois

What is High Assurance Software

```
"Testing shows the presence, not the
 absence, of bugs."
- E.W. Dijkstra
```

- Testing can't possibly cover the whole state space of complex systems
- Attack model isn't probablistic: assume that adversaries are hard at work to discover vulnerabilities.

galois

Properties to Assure

We're focusing on safety and security properties of embedded systems.

- Functional Correctness: a program always implements a given relation of inputs and outputs
- Memory Safety: each read and write to memory only accesses locations the programmer has explicitly allowed
- Timing Safety: real-time deadlines are always met, for any set of inputs

galois

Correct by Construction

- Methods exist for proving properties of existing programs - but they're often not complete or difficult to use
- Memory and type safety is very hard to prove about an existing program
- We made a new *Domain Specific Language* language where these properties are guaranteed *correct by construction*

galois

Domain Specific Language

What is a Domain Specific Language (DSL)?

- sed, awk: classic DSLs for text manipulation
- MATLAB: a DSL for matrix math
- Simulink: a DSL for composing MATLAB programs with streams

galois

The Ivory Language

Ivory is a Domain Specific Language

- Provable Memory and Type Safety
- Appropriate for embedded systems programming
- Interacts with legacy C

galois

The Ivory Language

Avoid the memory safety pitfalls of C:

- Unbounded array access
- Stack protection & allocation
- Null pointers
- Untyped pointers
- Heap allocation

galois

The Ivory Language

A restricted C is hard to program with.

- Hey, C is hard enough already!
- In C, the language does not provide good *abstraction*.
- End up using C's safety holes (untyped pointers, null pointers, unbounded arrays) to implement run-time behaviors.
- The C preprocessor is very limited for implementing compile-time abstractions.

galois

The Ivory Language

Ivory uses a full-featured programming language as a *macro language*

- Commonly called an *embedded domain-specific language* (EDSL).

We've embedded Ivory in the Haskell programming language

- Pure functional language
- Powerful type system
- General purpose

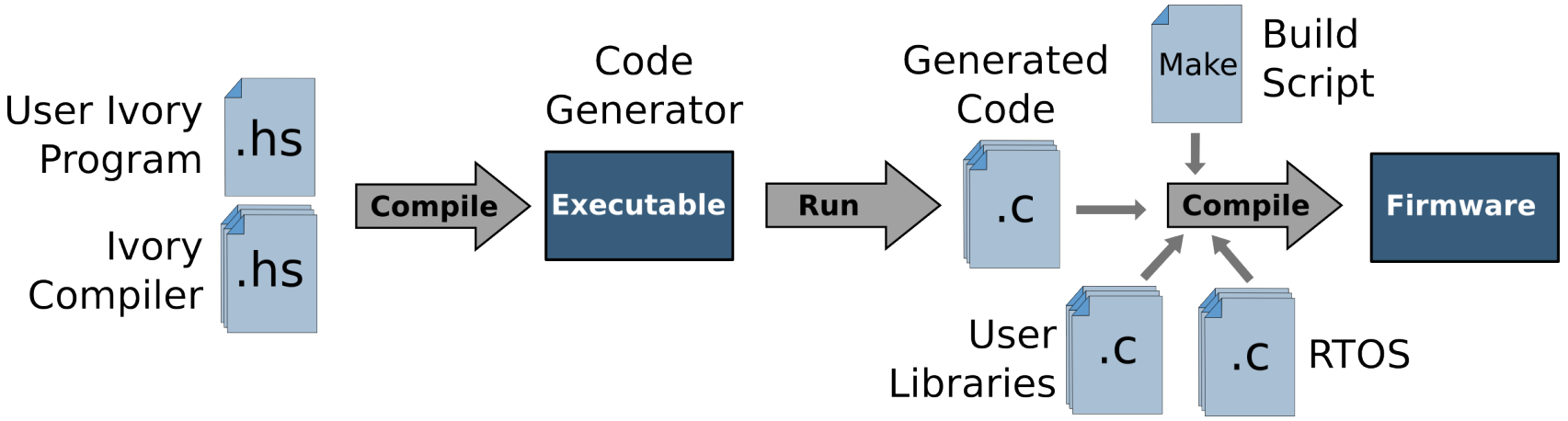
galois

Why Embedded Language?

- Building a programming language is hard!
- Get your programming language features for free:
  - Syntax & Parser
  - Lexical Scope
  - Type Checker
  - Macro Language
- Macro Language is Type Safe and Turing Complete

galois

Compiling and Running Ivory



galois

Fibonacci

```
arrayExample :: Def('[ Ref s (Array 4 (Stored Uint8))
                     , Uint8] :-> ())
arrayExample = proc "arrayExample" $
  \ arr x -> body $ do
    arrayMap $ \ ix -> do
      v <- deref (arr ! ix)
      store (arr ! ix) (v + x)
```

- Array length determines loop bound
- Limited loop combinators guarantee correct behavior

galois

Types, Structures, Arrays

```
struct foo
  { bar :: Stored IBool
  ; baz :: Stored IFloat
  ; qux :: Array 42 (Stored Uint32)
  }
```

galois

More Ivory Features

Built in features to enable and encourage safety:

- No implicit casting: `safeCast`, `castWith`, `castDefault`.
- Special facilities for dealing with bit data (e.g. memory-mapped peripherals)
- Automatic checks for bad behavior like divide by zero; user annotations with `assert` and `assume`.

galois

Program Composition

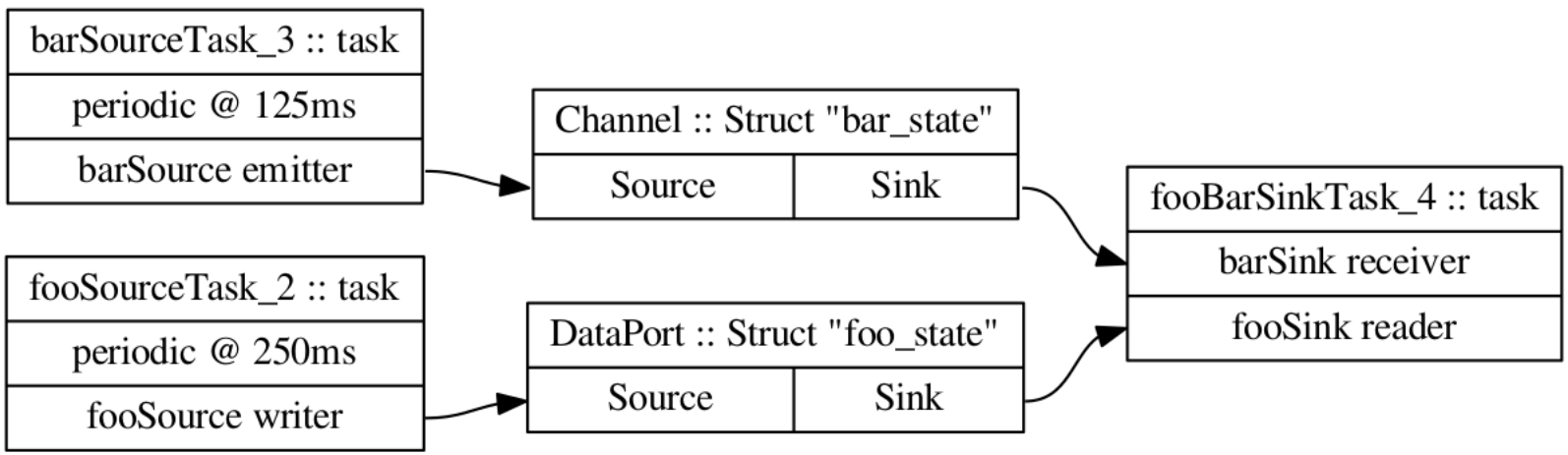
- Translating a "block diagram" program architecture into code is a classic software engineering problem.
  - How many LOC for ArduCopter's global state?
- RTOS facilities are tricky to use correctly
- Want guarantees about data flow, timing of our resulting composition

galois

Tower

Framework for composing Ivory programs with real-time tasks.

- Primitives are *global state* and *communication channels*
- Enforce memory and timing safety between tasks
- Generate safe glue code, graphs, and output for tools



galois

At last: SMACCMPilot

- Simple "stabilize mode" PID loops, RC control only
- MAVLink telemetry, parameters
- Uses the ArduPilot `AP_AHRS` library to run sensors and sensor fusion.
- PX4FMU 1.7 + PX4IOAR

galois

At last: SMACCMPilot

- Application is 2400 lines of Ivory code
  - ArduCopter subdir is 10k lines
- 5000 lines of Ivory generated for MAVLink
- 200 lines of Tower describe 9 tasks, 11 channels
- Under 9 man-months of effort

galois

SMACCMPilot System Diagram

(jump to pdf)

galois

At last: SMACCMPilot

Coming soon:

- Hardware Drivers
- Control Systems
- Networking

galois

SMACCMPilot.org

Ivory, Tower, and SMACCMPilot are open source.

`smaccmpilot.org`

galois

Questions?

galois