Supporting Software Sustainability With Lightweight Specifications

Mistral Contrastin, Matthew Danish,
Dominic Orchard*, Andrew Rice

UNIVERSITY OF CAMBRIDGE

University of Kent *

CamFort

Example: units-of-measure specifications

```
program energy
  != unit kg :: mass
  != unit m  :: height
  real :: mass = 3.00, gravity = 9.91, height = 4.20
  != unit kg m**2/s**2 :: potential_energy
  real :: potential_energy

  potential_energy = mass * gravity * height
end program energy
```

Example: units-of-measure specifications

```fortran
program energy
  != unit kg :: mass
  != unit m  :: height
  real :: mass = 3.00, gravity = 9.91, height = 4.20
  != unit kg m**2/s**2 :: potential_energy
  real :: potential_energy

  potential_energy = mass * gravity * height
end program energy
```

Check

```
$ camfort units-check energy1.f90

energy1.f90: Consistent. 4 variables checked.
```

Example: units-of-measure specifications

```fortran
program energy
  != unit kg :: mass
  != unit m  :: height
  real :: mass = 3.00, gravity = 9.91, height = 4.20
  != unit kg m**2/s**2 :: potential_energy
  real :: potential_energy

  potential_energy = mass * gravity * height
end program energy
```

Synthesise

```
$ camfort units-synth energy1.f90 energy1.f90

Synthesising units for energy1.f90
```

Example: units-of-measure specifications

```fortran
program energy
  != unit kg :: mass
  != unit m  :: height
  != unit m/s**2  :: gravity
  real :: mass = 3.00, gravity = 9.91, height = 4.20
  != unit kg m**2/s**2 :: potential_energy
  real :: potential_energy

  potential_energy = mass * gravity * height
end program energy
```

Synthesise

```
$ camfort units-synth energy1.f90 energy1.f90

Synthesising units for energy1.f90
```

CamFort

https://github.com/camfort/camfort/wiki

Currently:

- units-of-measure
- stencil specifications (shape of array access)

In progress:

- general boolean properties with auto-tests (QuickCheck)

Future?

- Get in touch and tell us what you need!!!