Safe Concurrent Abstractions for WSNs

Going Beyond Multi-threaded Programming

Francisco Sant'Anna Noemi Rodriguez Roberto Ierusalimschy
PUC-Rio, Brazil

Olaf Landsiedel Philippas Tsigas
Chalmers, Sweden

Motivation

Event-driven programming *(nesC, Contiki)*

→ **Multi-threaded programming** *(Protothreads, MantisOS)*

→ **Céu** **(our proposal)**

low level		high level
- unstructred - stackless	- independent threads - unrestricted shared memory	- composable threads - safety analysis

Céu Outlook

Our goal with Céu is to provide safe and expressive abstractions for programmers.

Key features

- compositions
  par/and par/or
- shared memory
- first-class timers
- internal events
- local scopes
- object system

**"Sense & broadcast every 10 min.
Force new sampling on external request."**

```
loop do
   par/or do
      _sensorRequest();
      int v = await SENSOR_READ;
      par/and do
         _radioBroadcast(&v);
         await RADIO_SENT;
      with
         await 10min;
      end
   with
      await RESTART;
   end
end
```

Static analysis

```
int v = 0;
par/or do
      await B;
      v = 1;
      await A;
      v = 2;
with
      await A;
      v = 3;
end
await A;
v = 4;
```

Initial Results

Evaluation

- 20-70% reduction in complexity
- no state variables
- less globals
- reasonable footprint

Application	Language	tokens	*Céu / nesC*	globals: states	globals: data	ROM	*Céu / nesC*	RAM	*Céu / nesC*
Trickle	nesC	477	0.32	2	2	3894	1.31	114	2.07
	Céu	155		0	0	5100		236	
DRIP	nesC	342	0.77	2	1	13296	1.08	415	1.27
	Céu	264		0	0	14424		525	
SRP	nesC	418	0.70	2	8	12266	1.18	1252	1.01
	Céu	291		0	4	14492		1261	
CTP	nesC	383	0.79	4	5	27712	1.07	3281	1.01
	Céu	303		0	2	29624		3327	
CC2420	nesC	590	0.76	1	2	12062	1.02	379	1.02
	Céu	447		0	0	12360		387	

CNPq PUC RIO CISB CENTRO DE PESQUISA E INOVAÇÃO SUECO-BRASILEIRO CHALMERS SAAB