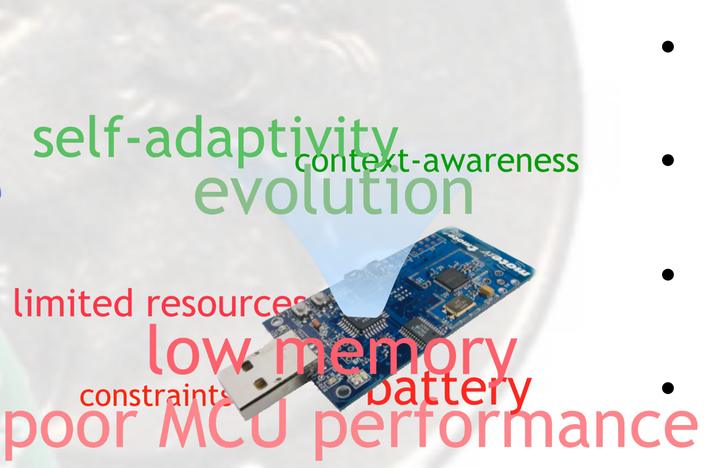
Towards Self-adaptive Software for Resource-constrained Cyberphysical Systems

Mikhail Afanasov*, Luca Mottola*†, Carlo Ghezzi*
*Politecnico di Milano (Italy), †Swedish Institute of Computer Science (SICS)
Contact email: **afanasov@elet.polimi.it**

Context-Oriented Programming for Cyberphysical Systems (CPSs)



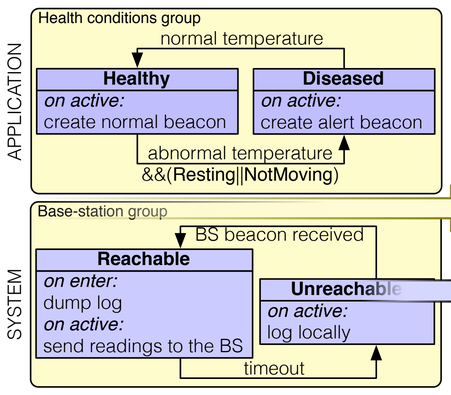
- CPSs need to be adaptable due to environmental dynamics
- Context is a representation of the environment
- Context-Oriented programming provides language abstractions for adaptation
- Context-Oriented programming is using context as a building block

Example: Wildlife Tracking

- Should send location data and health conditions
- Should transmit data if the base-station is in the reach
- Should locally log data in infrastructure-less situations

Context-Oriented nesC (ConesC)

Mapping




```
1 context group BaseStationG {
2  layered command void report(msg_t msg);
3 }implementation {
4  contexts Reachable,
5           Unreachable is default,
6           MyErrorC is error;
7  components Routing, Logging;
8  Reachable.Collection -> Routing;
9  Unreachable.DataStore -> Logging;}
```

```
1 context Unreachable {
2  transitions Reachable iff ActivityG.Running;
3  uses interface DataStore;
4 }implementation {
5  event void activated(){//...}
6  event void deactivated(){//...}
7  command bool check(){//...}
8  layered command void report(msg_t msg){
9   call DataStore.deposit(msg);}}
```

Example

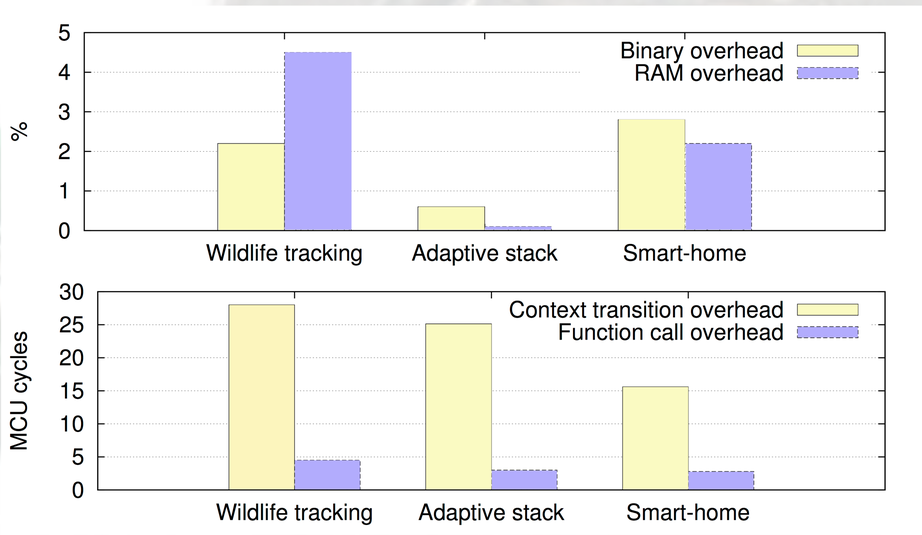
```
1 module User {
2  uses context group BaseStationG;
3 }implementation {
4  event void Timer.fired() {
5   call BaseStationG.report(msg);}
6  event void BaseStationG.contextChanged(context_t con) {
7   if(con == BaseStationG.Reachable) // DO SOMETHING...}}
```

```
1 module BaseStationContextManager {
2  uses context group BaseStationG;
3 }implementation {
4  event msg_t Beacon.receive(msg_t msg) {
5   activate BaseStationG.Reachable;
6   call BSReset.stop();
7   call BSReset.startOneShot(TIMEOUT);}
8  event void BSReset.fired() {
9   activate BaseStationG.Unreachable;}}
```

Applications are less complex*

Application	Average per-module: Variable declarations	Functions	Per-function states (avg)
Wildlife tracking – nesC	6	8	12567.3
Wildlife tracking – ConesC	3	2	6231.2
Smart-home controller – nesC	2	2	18654.2
Smart-home controller – ConesC	0,8	1,9	5678.3
Adaptive stack – nesC	2,5	3,25	9830.3
Adaptive stack – ConesC	0,4	1,6	3451.8

Overhead is **negligible***



*more details at Afanasov. M. et.al. “Context-Oriented Programming for Adaptive Wireless Sensor Network Software” DCOSS’14

Future Work

- Domain-specific model-checking
- Source-code generator
- Context-Oriented programming for other CPS platforms