Runtime Monitoring of Human Behaviour with Aggregate Computing on Android

VORTEX 2023

Volker Stolz[†], Giorgio Audrito[*]

[*]University of Turin, Italy
[†]Western Norway University of Applied Sciences, Bergen

@fm_volker@mastodon.social

July 18, 2023

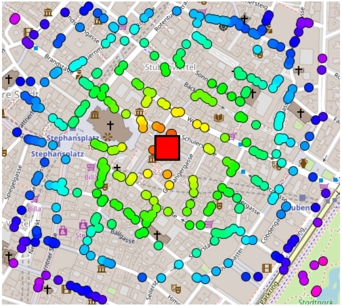
Overview

- Aggregate Programming in the field calculus with FCPP
- A temporal and spatial logics
- Applications & experiments:
  "evacuation", "friend-finding"
- Architecture of the Android application framework
  - → Bluetooth Low Energy (BLE) "Advertisement" and "Scanning"
  - → Reuse of code-base for simulation and deployment

Where can we use Aggregate Computing?

distance estimation, data summarisation (event detection),
selecting areas (network partitioning, channel establishment. . .),
inducing shapes (crowd dispersion, formation control. . .). . . and others!

Where can we use Aggregate Computing?



distance estimation, data summarisation (event detection),
selecting areas (network partitioning, channel establishment. . .),
inducing shapes (crowd dispersion, formation control. . .). . . and others!

Why are distributed systems hard to deal with?

diverse heterogeneous entities

- different computing power
- sensing and actuation capabilities

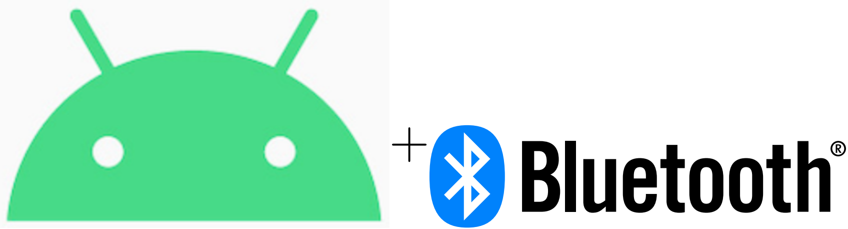
We need. . .

- device abstraction
- multi-platform frameworks
- not too bad so far. . .

Why are distributed systems hard to deal with?

diverse heterogeneous entities

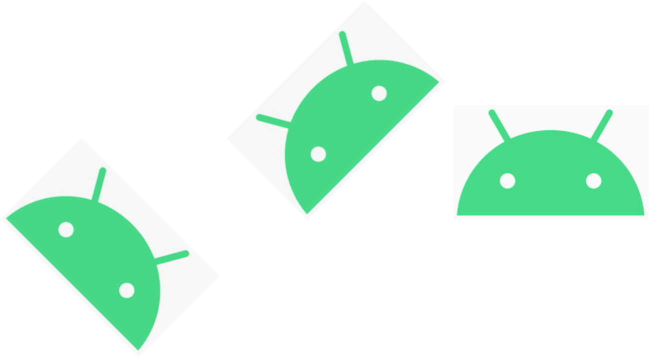
- different computing power
- sensing and actuation capabilities



Why are distributed systems hard to deal with?

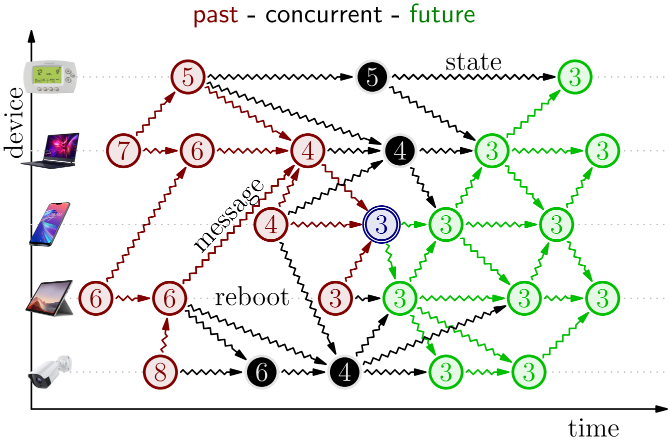
diverse heterogeneous entities

- different computing power — Still kind of true
- sensing and actuation capabilities — ditto



Formal model: Event structures

- a set of events E
- a DAG of messages $\rightsquigarrow$
- a causality partial order $<$ (transitive closure of $\rightsquigarrow$)



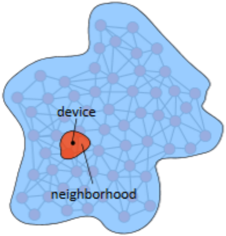
A Simple Concrete Computational Model

simplifying assumptions. . .

- the same program is executed in every event
- . . . can still execute different code through branching
- messages are sent through broadcast (can extend to pointwise messages)

Round:

1. gather data received, stored and sensed
2. compute the program
3. broadcast the result to neighbours
4. perform actuation as computed
5. receive messages while sleeping

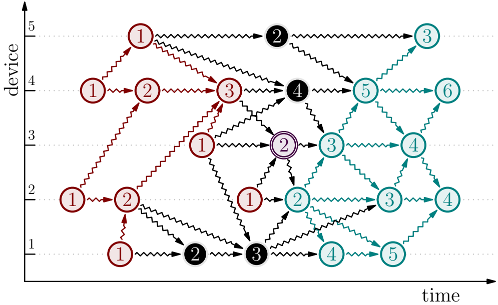


Principal Coordination Construct: nbr(e)

- represents interaction between neighbour devices
- sends result of e to neighbours (duality outgoing - incoming)
- collects neighbour's values for the same e into a neighbouring field

$$\boxed{\texttt{nbr}(\texttt{e}_c)}$$

neighbouring field of counters.



$\texttt{e}_c \longrightarrow 2$, broadcast 2

$\texttt{nbr}(\texttt{e}_c) \longrightarrow \phi$ where

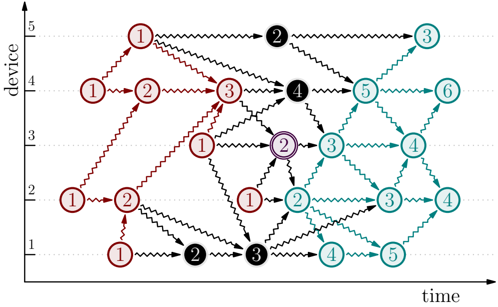
$\phi = \delta_2 \mapsto 1, \delta_3 \mapsto 2, \delta_4 \mapsto 3$

Principal Coordination Construct: `nbr(e)`

- represents interaction between neighbour devices
- sends result of e to neighbours (duality outgoing - incoming)
- collects neighbour's values for the same e into a neighbouring field

```
sum_hood(nbr(1))
```

counts the number of neighbours



Other functions on fields:

- `sum_hood`
- `min_hood`
- `all_hood`
- `any_hood`

Syntax & Semantics

Syntax of past-CTL and SLCS

$$\begin{array}{rll}
\phi ::= & \bot \mid \top \mid q \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \Rightarrow \phi) \mid (\phi \Leftrightarrow \phi) & \text{logical} \\
& \mid (\mathrm{P}\,\phi) \mid (\mathrm{AP}\,\phi) \mid (\mathrm{EP}\,\phi) \mid (\mathrm{H}\,\phi) \mid (\mathrm{AH}\,\phi) \mid (\mathrm{EH}\,\phi) & \text{temporal} \\
& \mid (\mathrm{Y}\,\phi) \mid (\mathrm{AY}\,\phi) \mid (\mathrm{EY}\,\phi) \mid (\phi\,\mathrm{S}\,\phi) \mid (\phi\,\mathrm{AS}\,\phi) \mid (\phi\,\mathrm{ES}\,\phi) & \\
& \mid (\Box\,\phi) \mid (\Diamond\,\phi) \mid (\partial\,\phi) \mid (\partial^{-}\,\phi) \mid (\partial^{+}\,\phi) & \text{spatial} \\
& \mid (\phi\,\mathcal{R}\,\phi) \mid (\phi\,\mathcal{T}\,\phi) \mid (\phi\,\mathcal{U}\,\phi) \mid (\mathcal{G}\,\phi) \mid (\mathcal{F}\,\phi) &
\end{array}$$

Temporal & spatial scope:

- $\mathrm{Y}\,\phi$: "ϕ held in the previous event on the same device";
- $\mathrm{EY}\,\phi$: "ϕ held in some previous event on any device";
- $\phi\,\mathrm{S}\,\psi$: "ψ held in some past event on the same device, and ϕ has held on the same device since then";
- $\phi\,\mathrm{AS}\,\psi$ (resp. $\phi\,\mathrm{ES}\,\psi$): "for all paths (resp. exists a path) of messages reaching the current event, ψ held in some event of the path and ϕ has held since then".

Syntax & Semantics

Syntax of past-CTL and SLCS

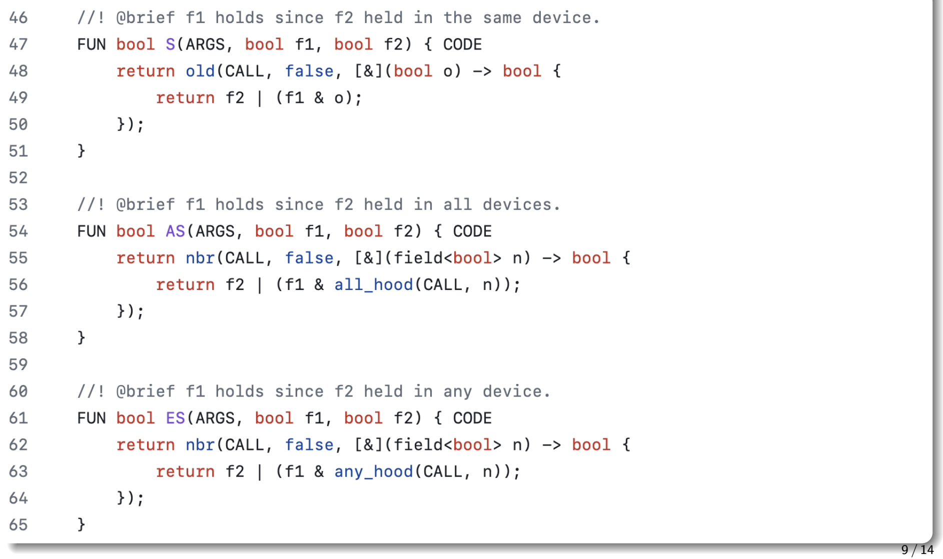
$$\begin{array}{rll}
\phi ::= & \bot \mid \top \mid q \mid (\neg\phi) \mid (\phi\wedge\phi) \mid (\phi\vee\phi) \mid (\phi\Rightarrow\phi) \mid (\phi\Leftrightarrow\phi) & \text{logical} \\
& \mid (\mathrm{P}\,\phi) \mid (\mathrm{AP}\,\phi) \mid (\mathrm{EP}\,\phi) \mid (\mathrm{H}\,\phi) \mid (\mathrm{AH}\,\phi) \mid (\mathrm{EH}\,\phi) & \text{temporal} \\
& \mid (\mathrm{Y}\,\phi) \mid (\mathrm{AY}\,\phi) \mid (\mathrm{EY}\,\phi) \mid (\phi\,\mathrm{S}\,\phi) \mid (\phi\,\mathrm{AS}\,\phi) \mid (\phi\,\mathrm{ES}\,\phi) & \\
& \mid (\Box\,\phi) \mid (\Diamond\,\phi) \mid (\partial\,\phi) \mid (\partial^-\,\phi) \mid (\partial^+\,\phi) & \text{spatial} \\
& \mid (\phi\,\mathcal{R}\,\phi) \mid (\phi\,\mathcal{T}\,\phi) \mid (\phi\,\mathcal{U}\,\phi) \mid (\mathcal{G}\,\phi) \mid (\mathcal{F}\,\phi) &
\end{array}$$

Temporal & spatial scope:

- $\Box\,\phi$ (interior): true at points where all neighbours satisfy ϕ;
- $\Diamond\,\phi$ (closure): true at points where a neighbour satisfies ϕ;
- ∂, ∂^- and ∂^+: *boundary* (closure without interior), *interior boundary* (set without the interior) and *closure boundary* (closure without the set).

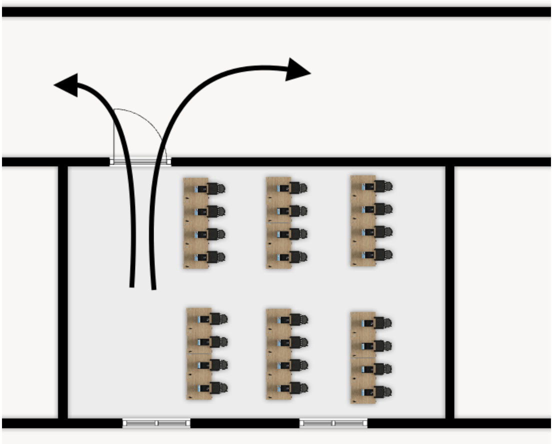
Runtime Monitors in FCPP

androidDemoApp/fcpp-android/lib/coordination/past_ctl.hpp



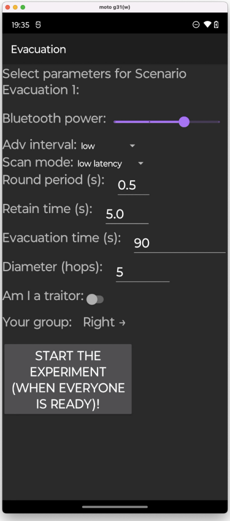
```
//! @brief f1 holds since f2 held in the same device.
FUN bool S(ARGS, bool f1, bool f2) { CODE
    return old(CALL, false, [&](bool o) -> bool {
        return f2 | (f1 & o);
    });
}

//! @brief f1 holds since f2 held in all devices.
FUN bool AS(ARGS, bool f1, bool f2) { CODE
    return nbr(CALL, false, [&](field<bool> n) -> bool {
        return f2 | (f1 & all_hood(CALL, n));
    });
}

//! @brief f1 holds since f2 held in any device.
FUN bool ES(ARGS, bool f1, bool f2) { CODE
    return nbr(CALL, false, [&](field<bool> n) -> bool {
        return f2 | (f1 & any_hood(CALL, n));
    });
}
```

Evacuation Experiment



- app partitions user into "left" or "right" group (randomly)
- on evacuation-begin, timer starts (manually)
- phone-display shows group-membership
- subjects evacuate according to their group
- expected outcome: app shows groups eventually correctly partitioned (or "traitor" detected)

Evacuation Experiment: UI



Select parameters for Scenario Evacuation 1:

Bluetooth power:

Adv interval: low

Scan mode: low latency

Round period (s): 0.5

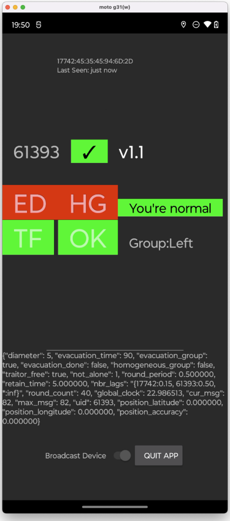
Retain time (s): 5.0

Evacuation time (s): 90

Diameter (hops): 5

Am I a traitor:

Your group: Right →



17742:45:35:45:94:6D:2D
Last Seen: just now

61393 ✓ v1.1

You're normal

Group:Left

{"diameter": 5, "evacuation_time": 90, "evacuation_group": true, "evacuation_done": false, "homogeneous_group": false, "traitor_free": true, "not_alone": 1, "round_period": 0.500000, "retain_time": 5.000000, "nbr_lags": "{17742:0.15, 61393:0.50, *:inf}", "round_count": 40, "global_clock": 22.986513, "cur_msg": 82, "max_msg": 82, "uid": 61393, "position_latitude": 0.000000, "position_longitude": 0.000000, "position_accuracy": 0.000000}

Some results:

- Works “well” with sub-second period.
- Visible load on battery.
- Additional “friend-finding” experiment (a la “hot & cold”) more challenging (flakyness, low N, UI/instruction issue)

Evacuation Experiment: Properties

- *ED*: "**E**vacuation **D**one"; time-limit reached.
- *L*: user is part of the "**l**eft" group, false otherwise.

- $\phi_{HG} = (L \Rightarrow \mathcal{G}\, L) \wedge (\neg L \Rightarrow \mathcal{G}\, \neg L)$: user is part of **h**omogeneous **g**roup.
- $\phi_{TF} = \mathrm{AH}(ED \Rightarrow \phi_{HG})$: "**t**raitors" **f**ound at end of experiment.

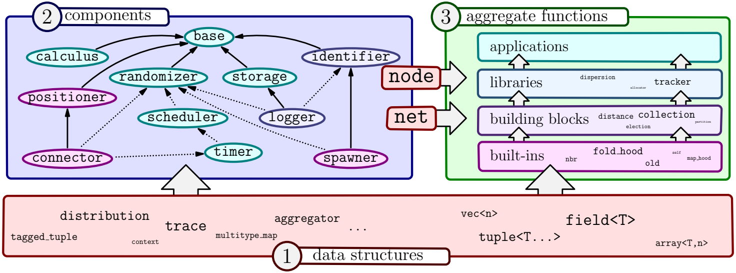
Operators:

- $\mathcal{G}\, \phi, \mathcal{F}\, \phi$ (everywhere, somewhere): true where ϕ holds in every (resp. some) point of every (resp. some) incoming path.
  Here: "If the user is part of the left group, then everyone in its connected area should also be in the left group; and similarly for the right group."
- $\mathrm{AH}(ED \Rightarrow \phi_{HG})$: "it has always and everywhere been the case that after the evacuation is done everyone is within an homogeneous group"

App Architecture

FCPP main features — https://github.com/fcpp/

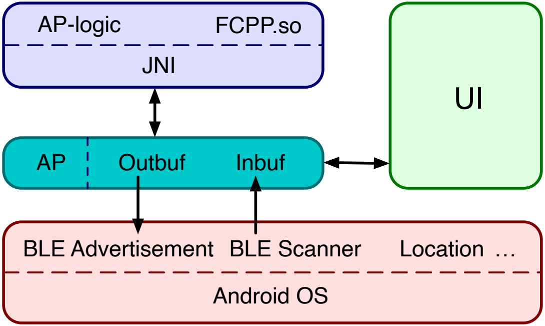
- C++ library used to develop distributed programs using it
  - manipulates C/C++ values
  - can use external C/C++ code
  - portable to any architecture with C++ compiler
- extensible component-based architecture
- runtime monitors for spatio-temporal properties on top of FCPP primitives



App Architecture

FCPP main features — https://github.com/fcpp/

- C++ library used to develop distributed programs using it
  - manipulates C/C++ values
  - can use external C/C++ code
  - portable to any architecture with C++ compiler
- Here: cross-compiled to Android architectures



Conclusion & Future Work

Conclusion

- Shown that https://github.com/fcpp portable & adaptable
- Discovered quite some variability in behaviour of Android phones
- Difficult to (globally) observe status of experiment through human proxies (even with central logging for debugging)

Future Work

- iOS-version, larger experiment, outdoors, . . .
  to fine-tune comms-parameters & energy-consumption.
- Close the gaps between design, simulation and deployment.
- Formalization around spatio-temporal properties and their equivalences.
- Find partner in application domain.

Conclusion & Future Work

Conclusion

- Shown that https://github.com/fcpp portable & adaptable
- Discovered quite some variability in behaviour of Android phones
- Difficult to (globally) observe status of experiment through human proxies (even with central logging for debugging)

Future Work

- iOS-version, larger experiment, outdoors, . . .
  to fine-tune comms-parameters & energy-consumption.
- Close the gaps between design, simulation and deployment.
- Formalization around spatio-temporal properties and their equivalences.
- Find partner in application domain.

Thank You!