Towards Aggregate Monitoring of Spatio-temporal Properties

Giorgio Audrito, Gianluca Torta

University of Torino, Italy

Verification and mOnitoring at Runtime EXecution (VORTEX) 2021

Online, 12 July 2021

Challenges: Programming for the IoT

Programming for the IoT poses several non-trivial challenges:

- diverse heterogeneous entities $\longrightarrow$ device abstraction?
- collaboration vs selfishness $\longrightarrow$ centralization? coordination?
- dynamic goals and environment $\longrightarrow$ adaptive algorithms? runtime verification?
- data security and privacy $\longrightarrow$ cryptography? localised aggregation?

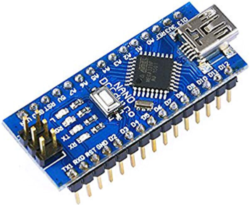
novel approach
aggregate computing · field calculus

Classical paradigms, algorithms and languages hardly deal with these expectations

Challenges: Runtime Verification for the IoT

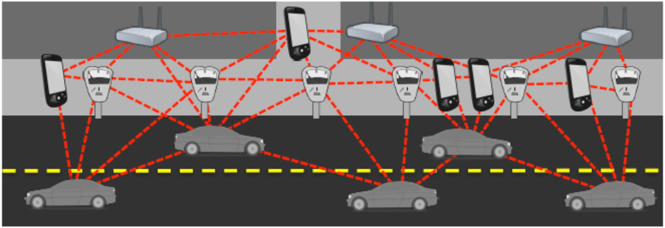
Several requirements need to be met:

- fully distributed monitors *(multi-hop networks)*
- monitors integrated within the IoT system
- dynamic devices and monitors *(may fail, join, move)*
- low resource consumption *(limited device capabilities)*



Outline

- recall the field calculus (FC), a programming language for the IoT
- recall the past-CTL temporal logic and its translation in FC
- recall the SLCS spatial logic and its translation in FC
- discuss sample spatio-temporal properties



Aggregate Computing*

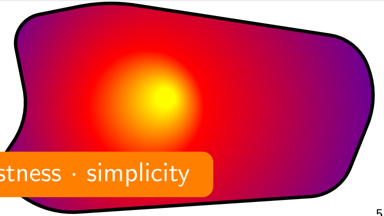
Shifting the viewpoint

- From single-device focus and query-based system programming
- To data-based aggregate viewpoint:
  - *devices in pervasive computing environment as single aggregate machine*
  - *overall dispersed localised data as a single object: a computational field*
  - *fields are kept updated at all times (no queries)*
  - *aggregate programs as composable global specifications, locally interpreted by devices*†

*[J Beal, D Pianini, M Viroli. *Aggregate Programming for the Internet of Things.* IEEE Computer 48(9), 2015. 10.1109/MC.2015.261]

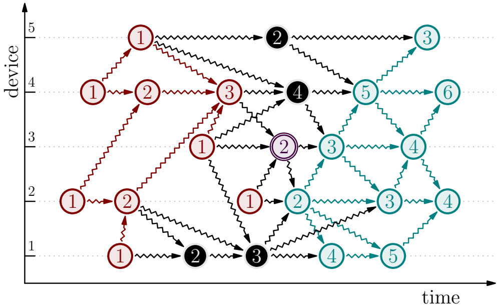
†[G Audrito, M Viroli, F Damiani, D Pianini, J Beal. *A Higher-order Calculus of Computational Fields.* ACM Transactions on Computational Logic 20(1), 2019. 10.1145/3285956]

$\longleftrightarrow$



adaptivity · resiliency · robustness · simplicity

Local Computational Model



Devices:

- are activated at periodic rounds
- communicate through broadcast

locally interpreting the given global specification

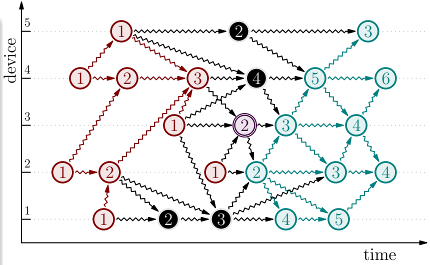
Round:

1. gather messages received, data stored and sensed
2. compute a same program (for every round and device)
3. broadcast the result to neighbours
4. perform actuation as computed
5. receive neighbours' messages while sleeping

Formal Language: Field Calculus

$$\texttt{e} ::= \texttt{x} \mid \phi \mid \texttt{c}(\overline{\texttt{e}}) \mid \texttt{b} \mid \texttt{d} \mid (\overline{\texttt{x}})\ \texttt{=>e} \mid \texttt{e}(\overline{\texttt{e}}) \mid \texttt{nbr}\{\texttt{e}\} \mid \texttt{rep}(\texttt{e}_1)\{\texttt{x=>e}_2\}$$

- core functional language
- two peculiar constructs
  - `nbr` for communication:
    · e is computed,
    · its value sent to neighbours,
    · values received from neighbours collected into a *neighbouring field value* ϕ.
  - `rep` for local state evolution:
    · e_1 initial value,
    · state is updated every round by e_2 substituting x for the previous value of the construct.



Past-Branching Time Logics (past-CTL)

$$\phi \ ::= \ \top \mid q \mid (\neg\phi) \mid (\phi \vee \phi) \mid (\mathrm{Y}\,\phi) \mid (\mathrm{EY}\,\phi) \mid (\phi\,\mathrm{S}\,\phi) \mid (\phi\,\mathrm{AS}\,\phi) \mid (\phi\,\mathrm{ES}\,\phi)$$

$$\mathrm{AY}\,\phi \triangleq \neg\,\mathrm{EY}\,\neg\phi \qquad \mathrm{P}\,\phi \triangleq \top\,\mathrm{S}\,\phi \qquad \mathrm{AP}\,\phi \triangleq \top\,\mathrm{AS}\,\phi \qquad \mathrm{EP}\,\phi \triangleq \top\,\mathrm{ES}\,\phi$$

$$\mathrm{H}\,\phi \triangleq \neg\,\mathrm{P}\,\neg\phi \qquad \mathrm{AH}\,\phi \triangleq \neg\,\mathrm{EP}\,\neg\phi \qquad \mathrm{EH}\,\phi \triangleq \neg\,\mathrm{AP}\,\neg\phi$$

- $\mathrm{Y}\,\phi$ (yesterday) when ϕ held in the previous event on the same device
- $\phi_1\,\mathrm{S}\,\phi_2$ (since) when ϕ_2 held in some past event and ϕ_1 has held since then
- $\mathrm{P}\,\phi$ (previously) when ϕ held in some past event
- $\mathrm{H}\,\phi$ (historically) when ϕ held in all past events

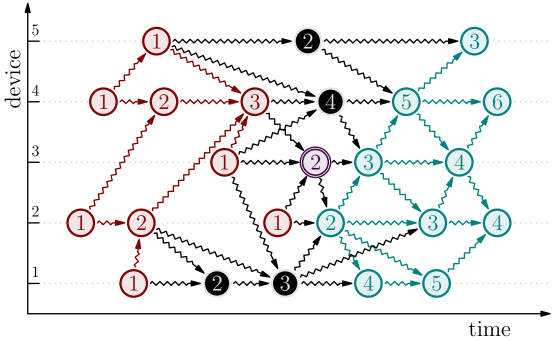
"event path" quantifiers A, E for all modalities:

- no quantifier $\longrightarrow$ *event path on the same device*
- $A \longrightarrow$ *all event paths to the current event*
- $E \longrightarrow$ *some event path to the current event*

5 modalities are fundamental, the rest is derived

Past-CTL on Event Structures

- $\mathrm{Y}\,\phi$, $\mathrm{EY}\,\phi$ just look at predecessor events
- $\mathrm{EP}\,\phi$ if ϕ holds in some red event
- $\mathrm{AH}\,\phi$ if ϕ holds in all red events
- $\mathrm{H}\,\phi$ if ϕ holds in all red events on device 3



Monitoring Past-CTL in Field Calculus

<table>
<tr><td>$[\![\top]\!]$</td><td>$=$</td><td><code>true</code></td><td>$[\![\phi_1 \vee \phi_2]\!]$</td><td>$=$</td><td>$[\![\phi_1]\!]$ <code>||</code> $[\![\phi_2]\!]$</td></tr>
<tr><td>$[\![\bot]\!]$</td><td>$=$</td><td><code>false</code></td><td>$[\![\phi_1 \wedge \phi_2]\!]$</td><td>$=$</td><td>$[\![\phi_1]\!]$ <code>&&</code> $[\![\phi_2]\!]$</td></tr>
<tr><td>$[\![q]\!]$</td><td>$=$</td><td><code>q()</code></td><td>$[\![\phi_1 \Rightarrow \phi_2]\!]$</td><td>$=$</td><td>$[\![\phi_1]\!]$ <code><=</code> $[\![\phi_2]\!]$</td></tr>
<tr><td>$[\![\neg\phi]\!]$</td><td>$=$</td><td><code>!</code>$[\![\phi]\!]$</td><td>$[\![\phi_1 \Leftrightarrow \phi_2]\!]$</td><td>$=$</td><td>$[\![\phi_1]\!]$ <code>==</code> $[\![\phi_2]\!]$</td></tr>
</table>

<table>
<tr><td>$[\![\mathrm{Y}\ \phi]\!]$</td><td>$=$</td><td><code>snd(share([false,false]){(old)=>[</code>$[\![\phi]\!]$<code>,locHood(fst(old))]})</code></td></tr>
<tr><td>$[\![\mathrm{AY}\ \phi]\!]$</td><td>$=$</td><td><code>snd(share([true, true]){(old)=>[</code>$[\![\phi]\!]$<code>,allHood(fst(old))]})</code></td></tr>
<tr><td>$[\![\mathrm{EY}\ \phi]\!]$</td><td>$=$</td><td><code>snd(share([false,false]){(old)=>[</code>$[\![\phi]\!]$<code>,anyHood(fst(old))]})</code></td></tr>
<tr><td>$[\![\phi_1 \mathrm{S}\ \phi_2]\!]$</td><td>$=$</td><td><code>share (false) {(old) =></code> $[\![\phi_2]\!]$ <code>||(</code>$[\![\phi_1]\!]$<code>&&locHood(old))}</code></td></tr>
<tr><td>$[\![\phi_1 \mathrm{AS}\ \phi_2]\!]$</td><td>$=$</td><td><code>share (false) {(old) =></code> $[\![\phi_2]\!]$ <code>||(</code>$[\![\phi_1]\!]$<code>&&allHood(old))}</code></td></tr>
<tr><td>$[\![\phi_1 \mathrm{ES}\ \phi_2]\!]$</td><td>$=$</td><td><code>share (false) {(old) =></code> $[\![\phi_2]\!]$ <code>||(</code>$[\![\phi_1]\!]$<code>&&anyHood(old))}</code></td></tr>
<tr><td>$[\![\mathrm{P}\ \phi]\!]$</td><td>$=$</td><td><code>share (false) {(old) =></code> $[\![\phi]\!]$ <code>||locHood(old)}</code></td></tr>
<tr><td>$[\![\mathrm{AP}\ \phi]\!]$</td><td>$=$</td><td><code>share (false) {(old) =></code> $[\![\phi]\!]$ <code>||allHood(old)}</code></td></tr>
<tr><td>$[\![\mathrm{EP}\ \phi]\!]$</td><td>$=$</td><td><code>share (false) {(old) =></code> $[\![\phi]\!]$ <code>||anyHood(old)}</code></td></tr>
<tr><td>$[\![\mathrm{H}\ \phi]\!]$</td><td>$=$</td><td><code>share (true) {(old) =></code> $[\![\phi]\!]$ <code>&&locHood(old)}</code></td></tr>
<tr><td>$[\![\mathrm{AH}\ \phi]\!]$</td><td>$=$</td><td><code>share (true) {(old) =></code> $[\![\phi]\!]$ <code>&&allHood(old)}</code></td></tr>
<tr><td>$[\![\mathrm{EH}\ \phi]\!]$</td><td>$=$</td><td><code>share (true) {(old) =></code> $[\![\phi]\!]$ <code>&&anyHood(old)}</code></td></tr>
</table>

Spatial Logic of Closure Spaces (SLCS)

$\phi ::= \top \mid q \mid (\neg\phi) \mid (\phi \vee \phi) \mid (\Diamond\, \phi) \mid (\phi\, \mathcal{R}\, \phi)$ fundamental op.

$\Box\, \phi \triangleq \neg(\Diamond(\neg\phi))$ $\partial\, \phi \triangleq (\Diamond\, \phi) \wedge \neg(\Box\, \phi)$ $\partial^-\, \phi \triangleq \phi \wedge \neg(\Box\, \phi)$ $\partial^+\, \phi \triangleq (\Diamond\, \phi) \wedge \neg\phi$

$\phi\, \mathcal{T}\, \psi \triangleq \phi\, \mathcal{R}(\Diamond\, \psi)$ $\phi\, \mathcal{U}\, \psi \triangleq \phi \wedge \Box\, \neg(\neg\psi\, \mathcal{R}\, \neg\phi)$ $\mathcal{F}\, \phi \triangleq \top\, \mathcal{R}\, \phi$ $\mathcal{G}\, \phi \triangleq \neg\, \mathcal{F}\, \neg\phi$

Local modalities

- $\Diamond\, \phi$ (closure) holds at points with some neighbour satisfying ϕ...

Global modalities

- $\phi\, \mathcal{R}\, \psi$ (reaches) holds at the start of paths satisfying ϕ ending in ψ...

two modalities are fundamental, the rest is derived
($\mathcal{R}$ chosen for presentation convenience)

SLCS Translation in Field Calculus

$\top$	`true`	$\phi_1 \vee \phi_2$	`F1 || F2`
$\bot$	`false`	$\phi_1 \wedge \phi_2$	`F1 && F2`
q	`q()`	$\phi_1 \Rightarrow \phi_2$	`F1 <= F2`
$\neg\phi$	`!F`	$\phi_1 \Leftrightarrow \phi_2$	`F1 == F2`
$\Box\,\phi$	`allHood(nbr{F})`	$\Diamond\,\phi$	`anyHood(nbr{F})`
$\phi_1\,\mathcal{R}\,\phi_2$	`if (F1) somewhere(F2) false`		

- `somewhere(F)` if `F` holds in some reachable device computing the function

```
def somewhere(F) { dist(F) < D }
```

  - `dist` is the hop-count optimal distance from closest device where `F` holds
    $\longrightarrow$ optimally $\neq$ exact: cannot know things instantaneously
  - `D` is the network diameter $\longrightarrow$ if closest `F` is farther, it doesn't exist

Spatio-temporal Monitoring

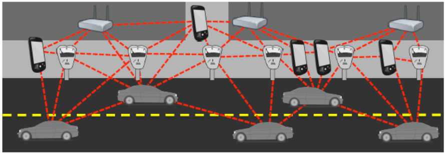
past-CTL vs SLCS

- incompatible interpretations: event structures vs graphs
  $\longrightarrow$ need to find a common ground
- past-CTL does capture some spatial properties
  $\longrightarrow$ path quantifiers A/E as everywhere/somewhere
- SLCS formulas have a temporal behaviour
  $\longrightarrow$ not captured by the graphs abstraction
- event structures can work as a common more general ground
  $\longrightarrow$ graph of the subjective present of events
- . . . where some overlap happens: $\Diamond \equiv \mathrm{EY}$, $\Box \equiv \mathrm{AY}$
  $\longrightarrow$ we view the immediate neighbourhood as in its immediate past
- but there is a strict expressiveness enhancement:
  $\longrightarrow$ past-CTL cannot talk about global present, while SLCS can

Motivating Examples

Network monitoring scenario

- atomic proposition s identifies servers
- d is true on devices (servers or not) which are down



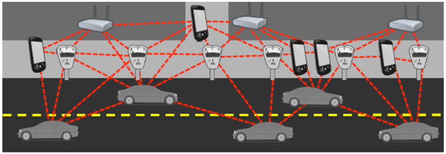
There is currently a server that has always been down

- $\mathrm{H}\, d \longrightarrow$ *the current device has always been down*
- $s \wedge \mathrm{H}\, d \longrightarrow$ *it is also a server*
- $\mathcal{F}(s \wedge \mathrm{H}\, d) \longrightarrow$ *there is currently such a server*

Motivating Examples

Network monitoring scenario

- atomic proposition s identifies servers
- d is true on devices (servers or not) which are down



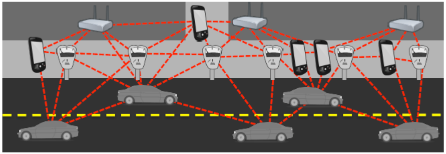
At some point in the past, every server was down

- $\mathcal{G}(s \Rightarrow d) \longrightarrow$ *everywhere there is a server, it must be down*
- $\mathrm{EP}\,\mathcal{G}(s \Rightarrow d) \longrightarrow$ *this formula has ever been true*

Motivating Examples

Network monitoring scenario

- atomic proposition s identifies servers
- d is true on devices (servers or not) which are down



Servers can always be reached through trustworthy devices

- $\neg\,\mathrm{P}\,d \longrightarrow$ *"trustworthy", i.e. never previously down*
- $(\neg\,\mathrm{P}\,d)\,\mathcal{R}\,s \longrightarrow$ *a server can be reached through those devices only*
- $\mathrm{AH}((\neg\,\mathrm{P}\,d)\,\mathcal{R}\,s) \longrightarrow$ *this property has always been satisfied*

Conclusions

- recall the field calculus (FC), a programming language for the IoT
- recall the past-CTL temporal logic and its translation in FC
- recall the SLCS spatial logic and its translation in FC
- discuss sample spatio-temporal properties

Future Work

- provide an abstract interpretation of SLCS formulas on event structures
- expand the set of supported modalities
- testing the approach on a simulated realistic case study

Thanks!