PRIN COMMON-WEARS
WP3 State of the Art (@UNITO)

(December 16, 2022)

Ferruccio Damiani (UNITO)

Task 3.1 Methodology

Aggregate Computing / Programming (AC/P)

Formal Methods and **Rigorous Software Engineering** perspective (when feasible)

Main ideas:

- **Programming a whole network and distributed data as a single entity**
- **Allow programming by functional composition, inheriting resilience properties**

Includes:

- **Language Design**
- **Algorithms**
- **Resilience properties (possibly “by-construction”)**
- **Distributed runtime verification**
- **Simulators and deployment platforms**

Language Design

Definition of a core calculus

- Syntax
- Operational semantics
- Type system
- Behavioural properties

$\mathtt{P}$	$::= \overline{\mathtt{F}}\ \mathtt{e}$	program
$\mathtt{F}$	$::= \mathtt{def}\ \mathtt{d}(\overline{\mathtt{x}})\ \{\mathtt{e}\}$	function declaration
$\mathtt{e}$	$::= \mathtt{x} \mid \mathtt{f}(\overline{\mathtt{e}}) \mid \mathtt{v} \mid \mathtt{if}(\mathtt{e})\{\mathtt{e}\}\mathtt{else}\{\mathtt{e}\} \mid \mathtt{nbr}\{\mathtt{e}\} \mid \mathtt{share}(\mathtt{e})\{(\mathtt{x}) \mathtt{=>} \mathtt{e}\}$	expression
$\mathtt{f}$	$::= \mathtt{d} \mid \mathtt{b}$	function name
$\mathtt{v}$	$::= \ell \mid \phi$	value
ℓ	$::= \mathtt{c}(\overline{\ell})$	local value
ϕ	$::= \overline{\delta} \mapsto \overline{\ell}$	neighbouring value

A couple of papers:

- G. Audrito, J. Beal, F. Damiani, D. Pianini, M. Viroli: ***Field-based Coordination with the Share Operator***. Log. Methods Comput. Sci. 16(4) (2020) https://doi.org/10.23638/LMCS-16(4:1)2020
- G. Audrito, R.Casadei, F. Damiani, G. Salvaneschi, M. Viroli. ***Functional Programming for Distributed Systems with XC***. ECOOP 2022: 20:1-20:28. https://doi.org/10.4230/LIPIcs.ECOOP.2022.20

Algorithms

- G. Audrito, F. Damiani, M. Viroli: ***Optimal Single-Path Information Propagation in Gradient-based Algorithms***. Science of Computer Programming 166, pp. 146–166, 2018. https://doi.org/10.1016/j.scico.2018.06.002
- G. Audrito, D. Pianini, F. Damiani, M. Viroli: ***Aggregate Centrality Measures for IoT-based Coordination***. Science of Computer Programming 203, 2021. https://doi.org/10.1016/j.scico.2020.102584
- G. Audrito, S. Bergamini, F. Damiani, M. Viroli: ***Effective Collective Summarisation of Distributed Data in Mobile Multi-Agent Systems***. AAMAS 2019: 1618-1626 https://doi.org/10.1145/3285956
- G. Audrito, R. Casadei, F. Damiani, D. Pianini, M. Viroli: ***Optimal resilient distributed data collection in mobile edge environments***. Comput. Electr. Eng. 96 (Part B) (2021) https://doi.org/10.1016/j.compeleceng.2021.107580
- Y. Mo, G. Audrito, S. Dasgupta, J. Beal: ***Near-Optimal Knowledge-Free Resilient Leader Election***. Automatica 146, 2022. https://doi.org/10.1016/j.automatica.2022.110583
- G. Aguzzi, G. Audrito, R. Casadei, F. Damiani, G. Torta, M. Viroli: ***A Field-based Computing Approach to Sensing-driven Clustering in Robot Swarms***. Swarm Intelligence, 2022 Sep 19:1-36. https://doi.org/10.1007/s11721-022-00215-y

Properties (possibly “by-construction”)

Self-stabilisation: behaviour is guaranteed to eventually attain a correct and stable final state despite any transient perturbation in state or topology

- M. Viroli, G. Audrito, J. Beal, F. Damiani, D. Pianini (2018). ***Engineering resilient collective adaptive systems by self-stabilisation***. ACM Transactions on Modeling and Computer Simulation, 28(2), 2018. https://doi.org/10.1145/3177774

Eventual consistency: behaviour converges to a final state that approximates a predictable limit, based on the continuous environment, as the density and speed of devices increases

- J. Beal, M. Viroli, D. Pianini, F. Damiani (2017). ***Self-adaptation to device distribution in the internet of things***. ACM Transactions on Autonomous and Adaptive Systems 12(3), 2017. https://doi.org/10.1145/3105758

Certified error bounds: linking the quality of the services to the amount of computing resources dedicated

- G. Audrito, F. Damiani, M. Viroli, E. Bini (2018). ***Distributed Real-Time Shortest-Paths Computations with the Field Calculus***. In: IEEE 39th Real-Time Systems Symposium (RTSS 2018). https://doi.org/10.1109/RTSS.2018.00013

Task 3.2 Verification Tools

Types of Verification

when performed

- static (with static analysis and **model checking**)
- at runtime (**Runtime Verification**)

type of properties

- spatial (**located** distributed systems)
- temporal

type of monitors

- centralized VS distributed
- hand-crafted VS automatically generated

type of spatially distributed system

- classic distributed system
- collective adaptive system (CAS)

Distributed Runtime Verification (I)

Runtime Verification (RV)

- observing a system at runtime

Important for engineering **Spatially Distributed Systems**:

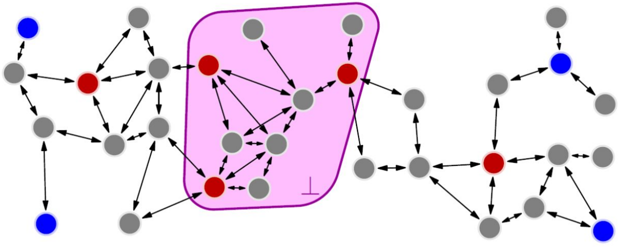
- monitors **spatially distributed** as well
- monitors **automatically generated** from formal specifications

Distributed Runtime Verification (II)

spatial properties

- D: true on devices in a **dangerous** area (TRUE in red points)
- R: true on devices in **recovery** points (TRUE in blue points)

$\neg D \, \mathcal{R} \, R$: true on devices that can reach a recovery point through devices in non-dangerous areas (FALSE in purple area)



STATE OF THE ART:

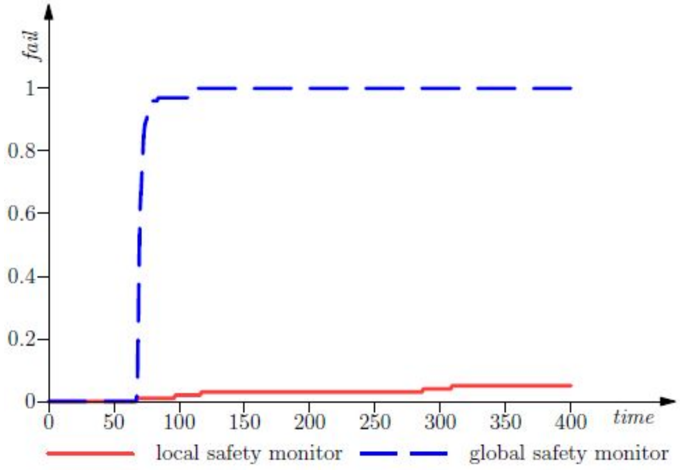
- Nenzi L, Bortolussi L., Ciancia V, Loreti M, Massink M, **Qualitative and quantitative monitoring of spatio-temporal properties with SSTL**. Logical Methods in Computer Science 14 (4) (2018)
- Audrito G, Casadei R, Damiani F, Stolz V, Viroli M. **Adaptive distributed monitors of spatial properties for cyber–physical systems**. Journal of Systems and Software. 2021 May 1;175: https://doi.org/10.1016/j.jss.2021.110908

Distributed Runtime Verification (III)

temporal properties

$$\varphi_1 := \mathrm{H}(\mathrm{Y}(safe \wedge alert) \rightarrow (safe \vee \neg alert)).$$

$$\varphi_2 := \mathrm{AH}(\mathrm{Y}(safe \wedge alert) \rightarrow (safe \vee \neg alert)).$$



STATE OF THE ART:

- Bauer, A., Falcone, Y. **Decentralized LTL monitoring.** Formal Methods in System Design 48 (1–2), 46–93 (2016)
- Audrito G, Damiani F, Stolz V, Torta G, Viroli M: **Distributed runtime verification by past-CTL and the field calculus**. Journal of Systems and Software. 2022 187: https://doi.org/10.1016/j.jss.2022.111251

Verification of Collective Adaptive Systems (CAS)

“A collective adaptive system consists of **multiple cooperating components** with **decentralised control**—these may be human or non-human— and **adapts itself** to unexpected problems arising from the environment in which it operates”

“(i) spontaneous, **dynamic network configuration**, with (ii) individual nodes acting in **parallel**, (iii) **constantly acting and reacting** to what the other agents are doing, and (iv) where the control tends to be highly **dispersed and decentralized** [...]”

STATE OF THE ART: Model Checking and Runtime Verification

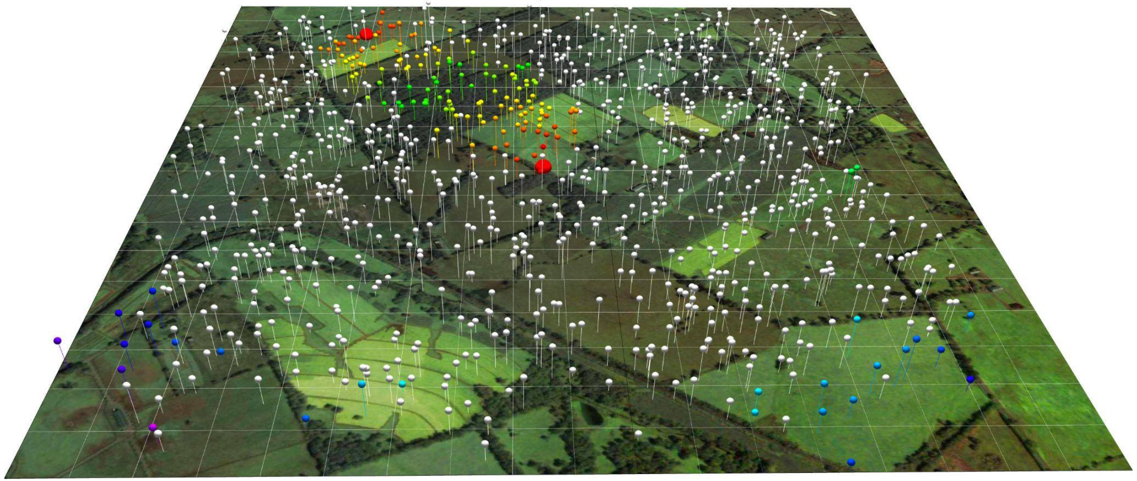
- Belmonte G, Ciancia V, Latella D, Massink M. From collective adaptive systems to human centric computation and back: spatial model checking for medical imaging. arXiv preprint arXiv:1607.02235 (2016)
- Ciancia V., Gilmore S., Grilletti G., Latella D., Loreti M. Massink, M. Spatio-temporal model checking of vehicular movement in public transport systems. STTT 20 (3), 289–311 (2018)
- Audrito G, Casadei R, Damiani F, Stolz V, Viroli M. Adaptive distributed monitors of spatial properties for cyber–physical systems. Journal of Systems and Software. 2021 May 1;175
- Audrito G, Damiani F, Stolz V, Torta G, Viroli M: Distributed runtime verification by past-CTL and the field calculus. Journal of Systems and Software. 2022 187

Task 3.3 Simulation

FCPP tool chain

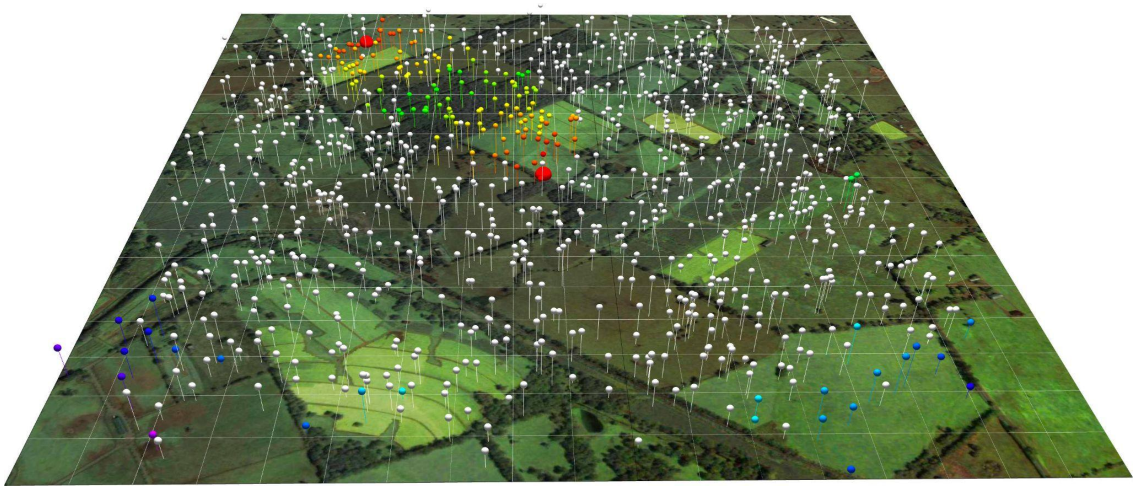
https://fcpp.github.io/

- **efficient, portable, small footprint**
- **integrated graphical simulator for development and debugging**
- **designed for deployment on real devices and various platforms**
    - **microcontrollers (Contiki, Miosix), smartphones (Android, work in progress), companion computers (Linux, planned), HPC platforms (work in progress)**



FCPP simulation features

- **interactive 3D graphical user interface**
- **programmable elements with aggregate support**
- **2D and 3D physics with acceleration and friction**
- **simplified stochastic wireless connection models**
- **random or regular geometric deployments**

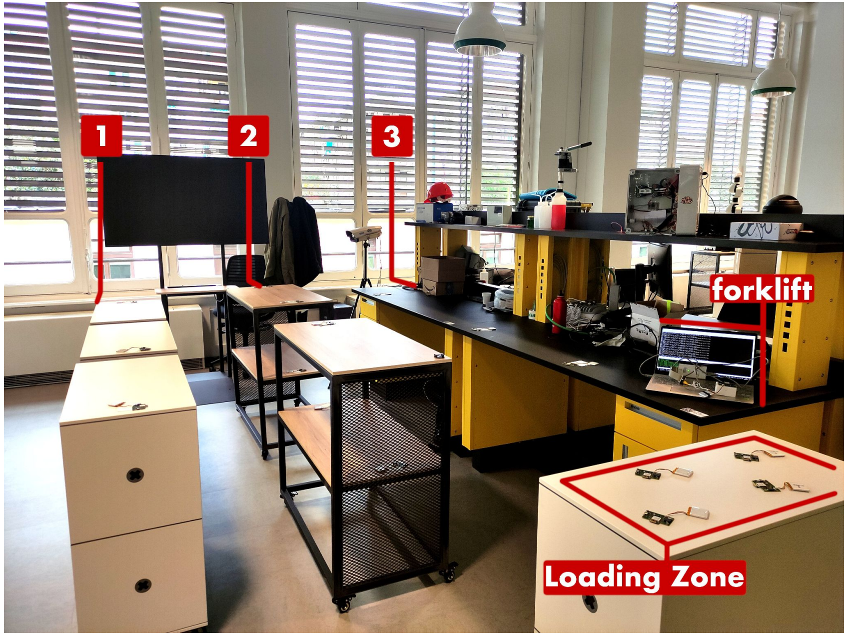


Task 2.3 Middleware

Industrial application scenario (with Concept Reply): warehouses, smart-buildings, autonomous delivery

Port of FCPP on a DecaWave Board, the DWM1001C module, which integrates:

- the Nordic Semiconductor nRF52832 general-purpose system on a chip (SoC), which offers:
  - a 64MHz ARM Cortex-M4 CPU with floating-point unit
  - a Bluetooth Low Energy (BLE) transceiver with Bluetooth 5 support
  - a 512 KB flash memory
  - a 64 KB RAM.
- the Decawave DW1000 Ultra Wideband (UWB) transceiver
- the STM LIS2DH12TR 3-axis accelerometer



Layout of the devices in the Proof-of-Concept deployment

Research infrastructure: “Area 42” lab (https://www.reply.com/it/area-42)

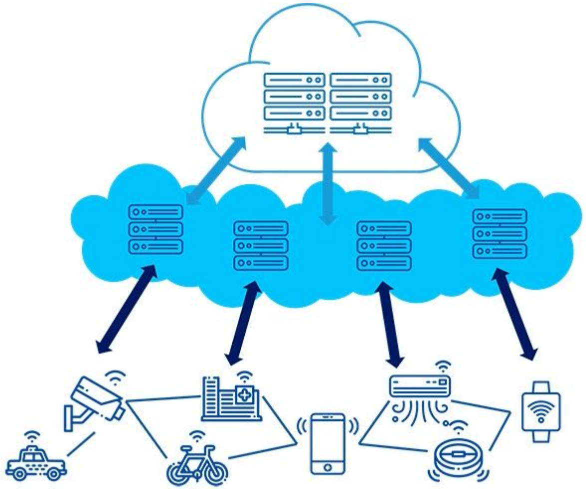
- Lorenzo Testa, Giorgio Audrito, Ferruccio Damiani, Gianluca Torta: Aggregate processes as distributed adaptive services for the Industrial Internet of Things. Pervasive Mob. Comput. 85: 101658 (2022)
- Audrito, G., Damiani, F., Rinaldi, S., Tagliabue, L.C., Testa, L., Torta, G. (2023). Aggregate Programming for Customized Building Management and Users Preference Implementation. In: Cicirelli, F., Guerrieri, A., Vinci, A., Spezzano, G. (eds) IoT Edge Solutions for Cognitive Buildings. Internet of Things. Springer

Towards Aggregate Computing on the Cloud

- Aggregate Computing currently implemented:
  - in simulators
  - in networks of hardware devices in pilot projects

- initial work towards implementation of AP into edge, cloud and device-edge-cloud continuum

- five-steps roadmap
  1. Data Processing Support → **first-cut implementation in FCPP**
  2. Multi-CPU Distribution
  3. Dynamic Multi-CPU Distribution
  4. Hybrid and Mirror Systems
  5. Dynamic Hybrid Systems



Audrito, G., Damiani, F., & Torta, G. (2022). Bringing aggregate programming towards the cloud. In International Symposium on Leveraging Applications of Formal Methods (pp. 301-317). Springer, Cham.