

Simulation of Field Calculus-based IoT Applications with Real-Time Guarantees

Giorgio Audrito*, Danilo Pianini†,
Ferruccio Damiani*, Mirko Viroli†, Enrico Bini*

* University of Torino, Italy

† University of Bologna, Italy

RTSS@Work 2018

39th IEEE Real-Time Systems Symposium

Nashville, Tennessee, USA, 11 December 2018



Raytheon
BBN Technologies

Motivation

The increasing availability of connected devices poses non-trivial challenges:

- diverse heterogeneous entities → device abstraction?
- collaboration vs selfishness → centralization? aggregation?
- dynamic goals and environment → adaptive algorithms?



Aggregate Computing

Directly programming the global behaviour in a simple way, while ensuring *adaptivity, robustness, error bounds and bounded convergence times*

Motivation

The increasing availability of connected devices poses non-trivial challenges:

- diverse heterogeneous entities → device abstraction?
- collaboration vs selfishness → centralization? aggregation?
- dynamic goals and environment → adaptive algorithms?



Aggregate Computing

Directly programming the global behaviour in a simple way, while ensuring *adaptivity, robustness, error bounds and bounded convergence times*

Motivation

The increasing availability of connected devices poses non-trivial challenges:

- diverse heterogeneous entities → device abstraction?
- collaboration vs selfishness → centralization? aggregation?
- dynamic goals and environment → adaptive algorithms?



Aggregate Computing

Directly programming the global behaviour in a simple way, while ensuring *adaptivity, robustness, error bounds and bounded convergence times*

Motivation

The increasing availability of connected devices poses non-trivial challenges:

- diverse heterogeneous entities → device abstraction?
- collaboration vs selfishness → centralization? aggregation?
- dynamic goals and environment → adaptive algorithms?



Aggregate Computing

Directly programming the **global behaviour** in a simple way, while ensuring *adaptivity, robustness, error bounds and bounded convergence times*

Large Distributed Networks

We may classify programming approaches in two groups:

	“query-based”	“data-based”
<i>approach</i>	computing when queries are issued	maintaining results updated
<i>resource usage</i>	on demand (best for fewer queries)	constant (best for frequent queries)
<i>responsiveness</i>	low: requires routing	high: data locally available
<i>quality</i>	high: fresh results	low: older results

Large Distributed Networks

We may classify programming approaches in two groups:

	“query-based”	“data-based”
<i>approach</i>	computing when queries are issued	maintaining results updated
<i>resource usage</i>	on demand (best for fewer queries)	constant (best for frequent queries)
<i>responsiveness</i>	low: requires routing	high: data locally available
<i>quality</i>	high: fresh results	low: older results

aggregate computing

Aggregate Computing Toolchain

1. Formal language, semantics and computational models for studying properties

$e ::= x \mid \phi \mid c(\bar{e}) \mid b \mid d \mid (\bar{x}) \Rightarrow e \mid e(\bar{e}) \mid \text{nbr}\{e\} \mid \text{rep}(e)\{x \Rightarrow e\}$

2. Language implementations (Java-based Protelis, etc.)

3. Network simulator (Alchemist) or real deployment (BBN Technologies)

Aggregate Computing Toolchain

1. Formal language, semantics and computational models for studying properties

$$e ::= x \mid \phi \mid c(\bar{e}) \mid b \mid d \mid (\bar{x}) \Rightarrow e \mid e(\bar{e}) \mid \text{nbr}\{e\} \mid \text{rep}(e)\{x \Rightarrow e\}$$

2. Language implementations (Java-based Protelis, etc.)

```
7@/**
8 * Classic Distance Estimation
9 *
10 * @param source whether the device is a source
11 * @return estimated distance
12 */
13@public def G_STD(source) {
14@  rep (old <- Infinity) {
15    mux (source) {0} else {minHood(old + self.nbrRange())}
16  }
17 }
```

3. Network simulator (Alchemist) or real deployment (BBN Technologies)

Aggregate Computing Toolchain

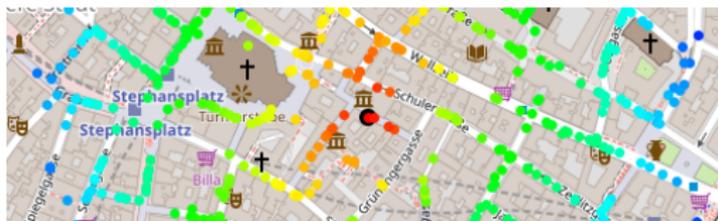
1. Formal language, semantics and computational models for studying properties

$$e ::= x \mid \phi \mid c(\bar{e}) \mid b \mid d \mid (\bar{x}) \Rightarrow e \mid e(\bar{e}) \mid \text{nbr}\{e\} \mid \text{rep}(e)\{x \Rightarrow e\}$$

2. Language implementations (Java-based Protelis, etc.)

```
7 @/**
8  * Classic Distance Estimation
9  *
10 * @param source whether the device is a source
11 * @return estimated distance
12 */
13 @public def G_STD(source) {
14 @   rep (old <- Infinity) {
15     mux (source) {0} else {minHood(old + self.nbrRange())}
16   }
17 }
```

3. Network simulator (Alchemist) or real deployment (BBN Technologies)



Demonstrations

	algorithm	scenario
1	classical distributed distance estimation	Vienna [A., D., V., B. Distributed Real-Time Shortest-Paths Computations with the Field Calculus, RTSS'18]
2	comparison of distance estimation algorithms	2000 × 200m corridor
3	Voronoi partitioning	Vienna
4	comparison of data collection algorithms	2000 × 200m corridor
5	stampede-prevention service	Vienna + real GPS traces

Thanks!

Demonstrations

	algorithm	scenario
1	classical distributed distance estimation	Vienna [A., D., V., B. Distributed Real-Time Shortest-Paths Computations with the Field Calculus, RTSS'18]
2	comparison of distance estimation algorithms	2000 × 200m corridor
3	Voronoi partitioning	Vienna
4	comparison of data collection algorithms	2000 × 200m corridor
5	stampede-prevention service	Vienna + real GPS traces

Thanks!

Demonstrations

	algorithm	scenario
1	classical distributed distance estimation	Vienna [A., D., V., B. Distributed Real-Time Shortest-Paths Computations with the Field Calculus, RTSS'18]
2	comparison of distance estimation algorithms	2000 × 200m corridor
3	Voronoi partitioning	Vienna
4	comparison of data collection algorithms	2000 × 200m corridor
5	stampede-prevention service	Vienna + real GPS traces

Thanks!

Demonstrations

	algorithm	scenario
1	classical distributed distance estimation	Vienna [A., D., V., B. Distributed Real-Time Shortest-Paths Computations with the Field Calculus, RTSS'18]
2	comparison of distance estimation algorithms	2000 × 200m corridor
3	Voronoi partitioning	Vienna
4	comparison of data collection algorithms	2000 × 200m corridor
5	stampede-prevention service	Vienna + real GPS traces

Thanks!

Demonstrations

	algorithm	scenario
1	classical distributed distance estimation	Vienna [A., D., V., B. Distributed Real-Time Shortest-Paths Computations with the Field Calculus, RTSS'18]
2	comparison of distance estimation algorithms	2000 × 200m corridor
3	Voronoi partitioning	Vienna
4	comparison of data collection algorithms	2000 × 200m corridor
5	stampede-prevention service	Vienna + real GPS traces

Thanks!

Demonstrations

	algorithm	scenario
1	classical distributed distance estimation	Vienna [A., D., V., B. Distributed Real-Time Shortest-Paths Computations with the Field Calculus, RTSS'18]
2	comparison of distance estimation algorithms	2000 × 200m corridor
3	Voronoi partitioning	Vienna
4	comparison of data collection algorithms	2000 × 200m corridor
5	stampede-prevention service	Vienna + real GPS traces

Thanks!