Ensuring Safe Composition of Distributed Processes

Jacob Beal

QA4SASO Workshop
@ IEEE SASO
September, 2015
A generative approach to safety

Restrict your development environment...

... to contain only resilient distributed systems.
Everything is a wireless computer
Example: Services for Mass Events
Example: Managing Crowd Danger
Distributed services are complex

How to make engineering resilience tractable?
Aggregate Programming

Aggregate Programming for the Internet of Things

Jacob Beal, Danilo Pisanelli, and Mirco Virola

Design and Deployment of an IoT Application-Oriented Testbed

Laura Bello, Simone Cران, Luca Davoli, Andrea Sorrell, Mirok Mancio, Marco Picone, and Gianluigi Femia

Repurposing Web Analytics to Support the IoT

Matteo Malgoso, Sarah Pianco, Rachel Jones, Mike Warding, Christopher Winstonley, and Nigel Davies
Aggregate Programming Stack
Field Calculus

Variable | Literal Value | Local Ops
---|---|---

Abstract & Call Functions

State

Communication

Domain Change

expression

special constructs

variable or value

function

program

\[
e ::= x \mid l \mid (b \, e) \mid (f \, e) \\
\mid (\text{rep} \, x \, w \, e) \mid (\text{nbr} \, e) \mid (\text{if} \, e \, e \, e)
\]

\[
w ::= x \mid l
\]

\[
F ::= \text{def} \, f(\overline{x}) \, e
\]

\[
P ::= \overline{F} \, e
\]

[Viroli et al., ‘13, Damiani et al., ‘15]
Field Calculus is Space-Time Universal

Space-time Universal = arbitrarily good approximation of any causal, finitely-approximable computation

[Beal, ’10; Beal et al., SCW 14]
Instantiation: Protelis

- Java-hosted & integrated
- Java-like syntax
- Eclipse support

```python
def distanceTo(source):
    rep(d <- Infinity) {
        mux (source) { 0 }
        else { minHood(nbr{d} + nbrRange) }
    }
```

Architecture:
Aggregate Programming Stack

- **Crowd Management**
  - dangerousDensity
  - crowdWarning
  - crowdTracking
  - safeDispersal

- **Collective Behavior**
  - collectivePerception
  - collectiveSummary
  - managementRegions

- **Perception**
  - summarize
  - average
  - regionMax

- **Action**
  - distanceTo
  - broadcast
  - partition

- **State**
  - timer
  - lowpass
  - recentTrue

- **Resilient Coordination Operators**
  - C
  - G
  - T
  - if

- **Field Calculus Constructs**
  - nбр
  - rep
  - if

- **Device Capabilities**
  - sensors
  - actuators
  - local functions
  - communication
  - state
  - restriction

**Application Code**

**Developer APIs**
Self-Stabilizing Building Blocks

Information spreading  Information collection  Short-term memory

Resilience by construction: all programs from these building blocks are also self-stabilizing!
Applying building blocks:

Example API algorithms from building blocks:

- distance-to (source)
- broadcast (source value)
- summarize (sink accumulate local null)
- integral (sink value)
- timer (length)
- random-voronoi (grain metric)
- broadcast-region (region source value)
- distance-avoiding-obstacles (source obstacles)
- max-likelihood (source p)
- path-forecast (source obstacle)
- average (sink value)
- region-max (sink value)
- limited-memory (value timeout)
- group-size (region)
- recent-event (event timeout)

Since based on these building blocks, all programs built this way are self-stabilizing!
def crowd-tracking (p) ;; Consider only Fruin LoS E or F within last minute ( if (recently-true (> (density-est p) 1.08) 60) ;; Break into randomized “cells” and estimate danger of each (+ 1 (dangerous-density (sparse-partition 30) p)) 0))

(def recently-true (state memory-time) ;; Make sure first state is false, not true... (rt-sub (not (T l l)) state memory-time))
(def rt-sub (started s m) (if state 1 (limited-memory s m)))

(def dangerous-density (partition p) ;; Only dangerous if above critical density threshold... (and (> (average partition (density-est p)) 2.17) ;; ... and also involving many people. (> (summarize partition + (/ 1 p) 0) 300)))

(def crowd-warning (p range) (> (distance-to (= (crowd-tracking p) 2)) range))

(def safe-navigation (destination p) (distance-avoiding-obstacles destination (crowd-warning p)))
Optimization of Dynamics

See our talk in the main program of SASO!
Predicting and Controlling Dynamics

Distance estimate: prediction from transient response:

\[ \Delta^+[t] \rightarrow \frac{d\Delta^+[t]}{dt} \cdot \text{diameter} \]

See our talk in the FOCAS/SCOPES workshop!
Aggregate Programming Stack
Crowd Safety Services

def dangerousDensity(p, r):
    let mr = managementRegions(r*2, () -> { nbrRange });
    let danger = average(mr, densityEst(p, r)) > 2.17 &&
        summarize(mr, sum, 1 / p, 0) > 300;
    if(danger) { high } else { low }

def crowdTracking(p, r, t):
    let crowdRgn = recentTrue(densityEst(p, r)>1.08, t);
    if(crowdRgn) { dangerousDensity(p, r) } else { none }

def crowdWarning(p, r, warn, t) {
    distanceTo(crowdTracking(p,r,t) == high) < warn
}
Dependency-Directed Recovery

Identify & shut down affected services

Non-dependent services still run

Dramatically better recovery time

Fewer services disrupted

See our talk in the main program of SASO!
Distributed Rewind and Replay
Tactical Cloud Services
Summary: Aggregate Methodology

Crowd Management
- dangerousDensity
- crowdWarning
- crowdTracking
- safeDispersal

Collective Behavior
- collectivePerception
- collectiveSummary
- managementRegions
- ...

Perception
- summarize
- average
- regionMax
- ...

Action
- distanceTo
- broadcast
- partition
- ...

State
- timer
- lowpass
- recentTrue
- ...

Application Code

Developer APIs

Resilient Coordination Operators

Field Calculus Constructs

Device Capabilities

Collective Behavior Constructs

Sensors & Local Functions
- communication
- state
- restriction

Actuators & Device Capabilities
- restricted

Built-in

Collective Behavior


Protelis' Program
- Enterprise Server
- Protelis' Daemon
- Protelis' Interpreter
- Protelis' Manager
- Protelis' Enterprise Variables
- Protelis' Protocols
- Protocols' Device
- Protocols' Service
Acknowledgements

- Shane Clark
- Partha Pal
- Kyle Usbeck
- Mirko Viroli
- Danilo Pianini
- Soura Dasgupta
- Raghu Mudumbai
- Amy Kumar
- Ferruccio Damiani