An Agent Framework for Agent Societies

Kyle Usbeck Jacob Beal [1]

Raytheon
BBN Technologies
10 Moulton St. Cambridge, MA

kusbeck@bbn.com jakebeal@bbn.com

Agents and Actors Reloaded (AGERE) Workshop at SPLASH 2011

October 24, 2011

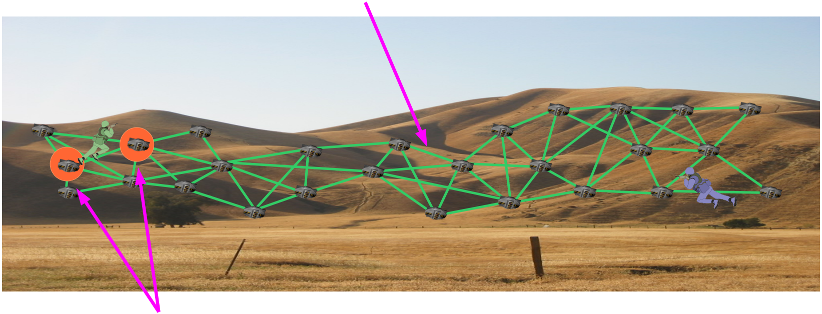
[1] Work sponsored by DARPA DSO under contract W91CRB-11-C-0052; the views and conclusions contained in this document are those of the authors and not DARPA or the U.S. Government.

Problem

Agent framework key weakness: controlling the global (emergent) behavior of a Multi-Agent System (MAS).

Problem

Agent framework key weakness: controlling the global (emergent) behavior of a Multi-Agent System (MAS).

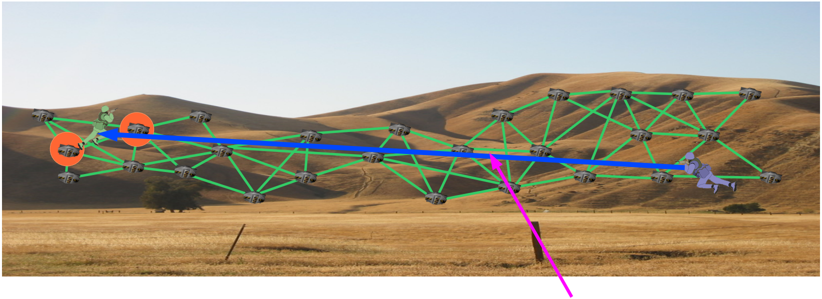


Intrusion Detection Sensor Network

Intrusion Detected

Problem

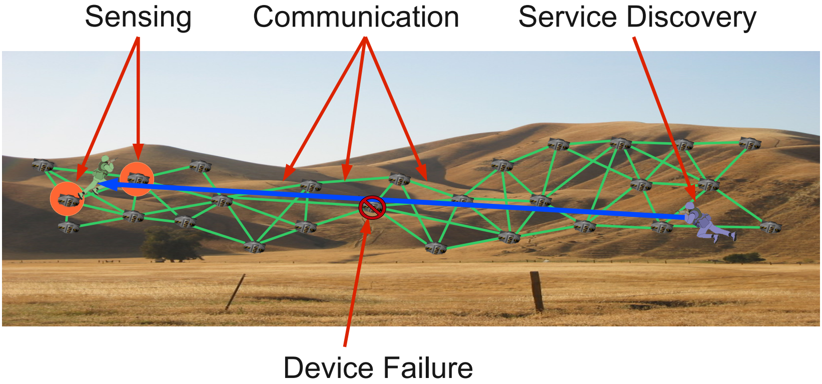
Agent framework key weakness: controlling the global (emergent) behavior of a Multi-Agent System (MAS).



Tracking Vector

Problem

Agent framework key weakness: controlling the global (emergent) behavior of a Multi-Agent System (MAS).



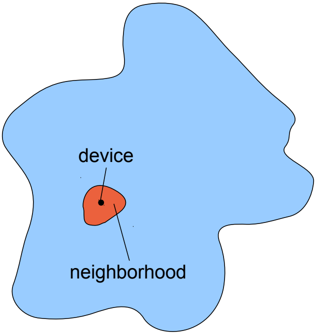
Sensing

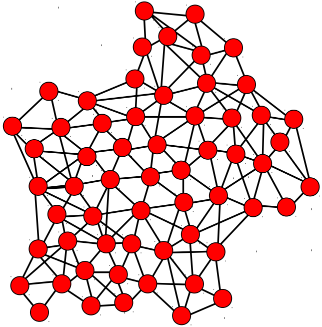
Communication

Service Discovery

Device Failure

Proto's Continuous Model





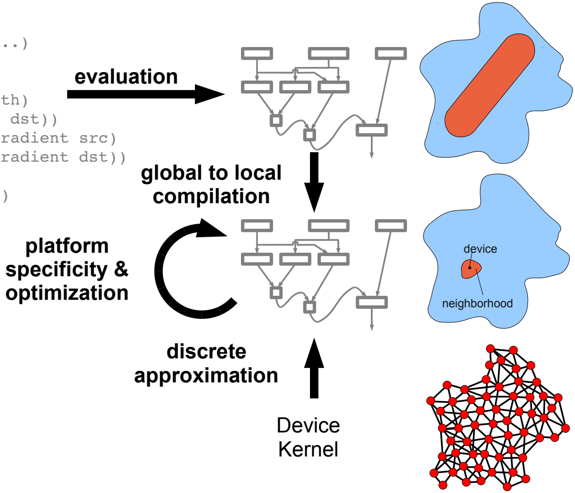
- Continuous space-time
- Infinite number of devices
- Neighbors' past state

- Approximate with discrete network of devices
- Signals transmit state

Benefits: simple, scalable, robust, adaptive

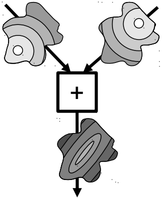
Global to Local Transformation

```
(def gradient (src) ...)
(def distance (src dst) ...)
(def dilate (src n)
  (<= (gradient src) n))
(def channel (src dst width)
  (let* ((d (distance src dst))
         (trail (<= (+ (gradient src)
                       (gradient dst))
                    d)))
    (dilate trail width)))
```

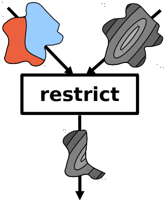


Proto’s Families of Primitives

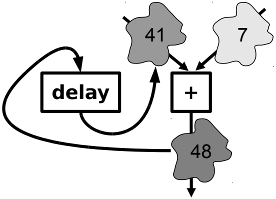
Pointwise



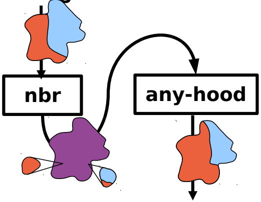
Restriction



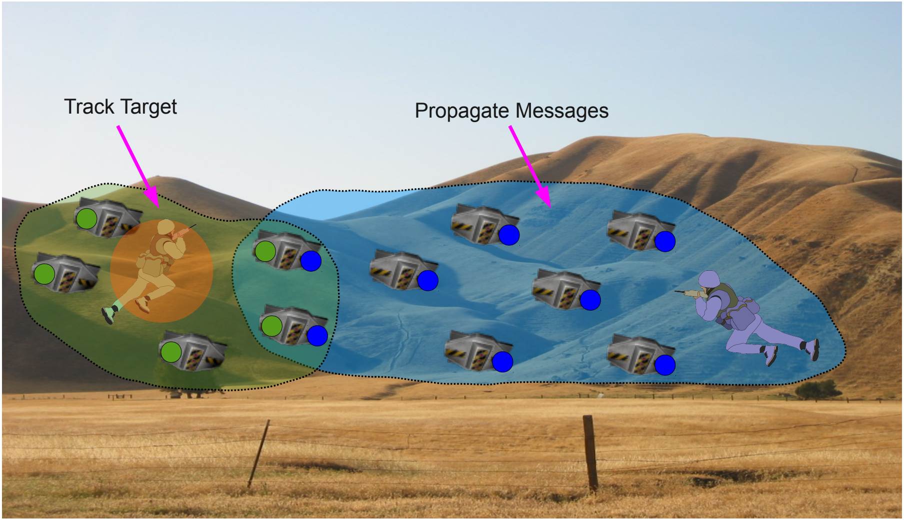
Feedback



Neighborhood



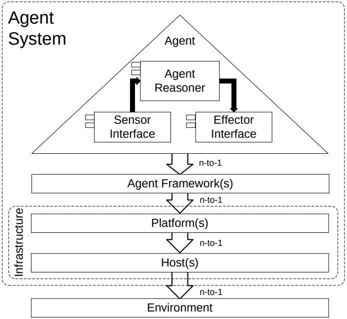
Mapping Agents to Proto



Agent System Reference Model/Architecture

ASRM defines seven *functional concepts* for agent systems:

- □ Agent Administration
- □ Directory Services
- □ Security and Survivability
- □ Messaging
- □ Mobility
- □ Conflict Management
- □ Logging

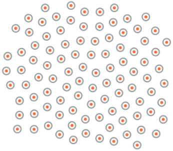


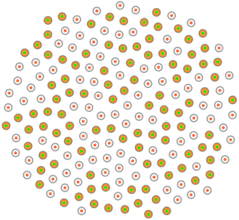
ASRA defines *architectural paradigms* for each functional concept.

Demo

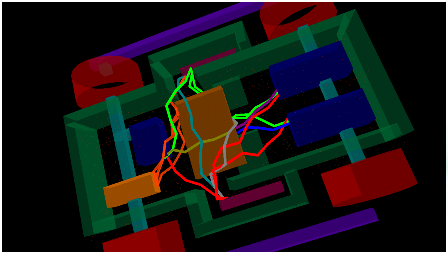
Current Progress on Functional Concepts

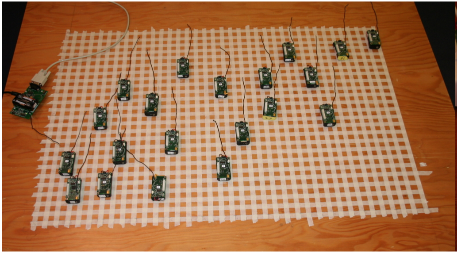
- ☑ Agent Administration
- ☑ Directory Services
- ⧅ Security and Survivability
- ☑ Messaging
- ☑ Mobility
- ⧅ Conflict Management
- ☒ Logging

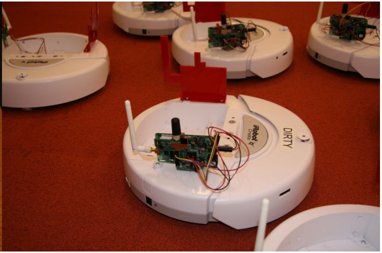




Framework Implementations







Contributions

Analysis of Proto's support for agent framework functional concepts.

Open Research Challenges:

- Conflict Management, Voting
- Security, Non-cooperative Agents
- Logging

`http://proto.bbn.com`

Thanks

Morphogenetically Assisted Design Variation (MADV) Team:

Raytheon
BBN Technologies

- **Jacob Beal (PI)**
- Aaron Adler (co-PI)
- Susan Katz (PM)
- Brett Benyo
- Taylor Campbell
- Jeff Cleveland
- Jessica Lowell
- Katie McGuire
- Hala Mostafa
- **Kyle Usbeck**
- Fusun Yaman

iRobot

- Annan Mozeika
- Gretchen Markiewicz

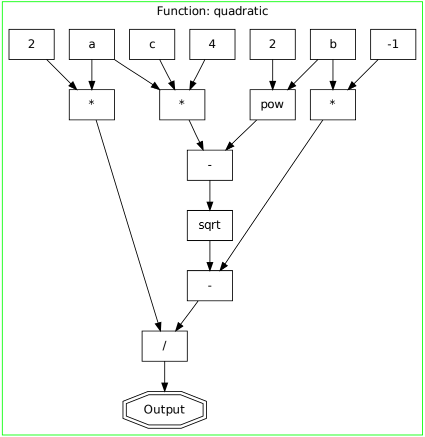
`http://madv.bbn.com`

Sponsored by DARPA

BACKUP

Dataflow Graph

```
(def quadratic (a b c)
   (/ (- (neg b)
         (sqrt (- (pow b 2)
                  (* 4 a c))))
      (* 2 a)))
```

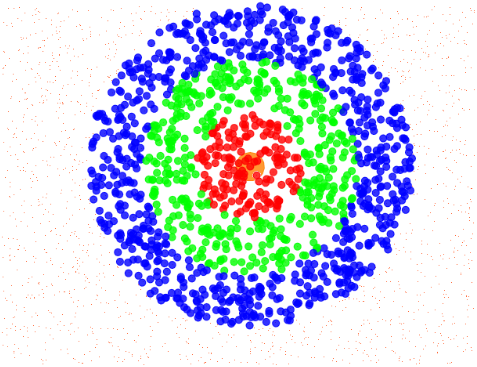


Back to Proto

Agent Framework Bullseye

Required components:

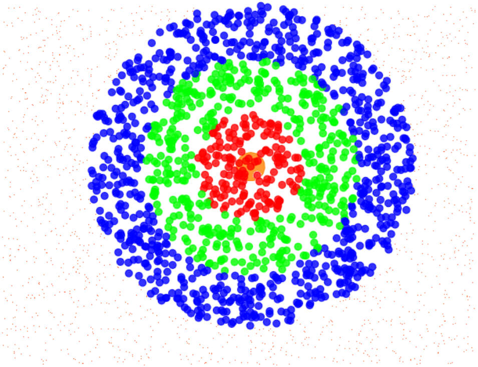
- Sensor
- Service Discovery
- Localization
- Communication



Back to Problem

Proto Bullseye

```
(def bullseye (src)
 (let ((d (distance-to src)))
  (if (< d 15) (red 1)
   (if (< d 30) (green 1)
    (if (< d 45) (blue 1)
     0)))))
(bullseye (sense 1))
```



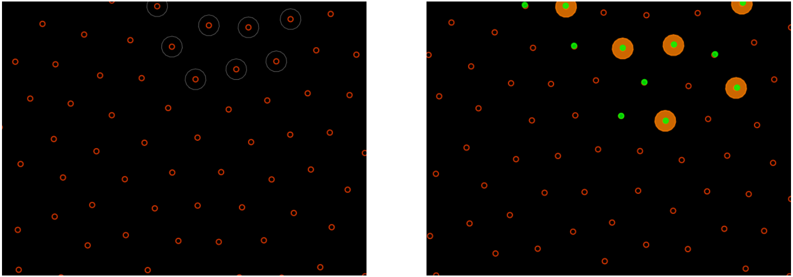
```
proto -s 0.1 -r 8 -n 1000 -m -l "(mov (all (bullseye (sense 1)) (brownian))))"
```

Back to Problem

Agent Administration

Includes:

- Instantiating agents
- Terminating agents
- Inspecting agent state



Example: Cellular-level scaling via replication

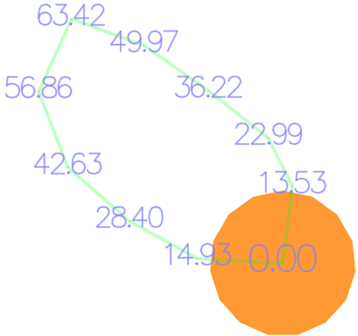
Directory Services

Enables locating and accessing shared resources (i.e., UDDI).

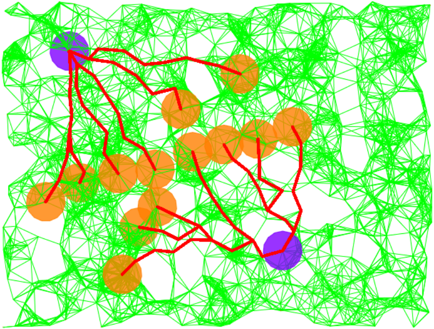
```
(distance-to (sense 1))
```

Note

(sense 1) is an operator that returns the location of a test sensor shown in orange.



Directory Services

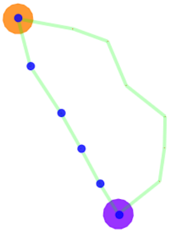


Example: Connecting data sources to data sinks

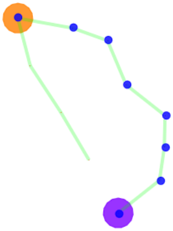
```
(connect (sense 1) (sense 2))
```

Security and Survivability

"Remain useful/dependable in the face of malice, error, or accident."



Before Disruption



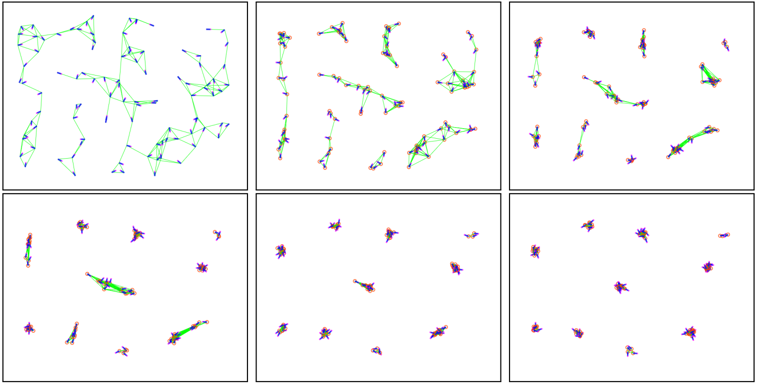
After Disruption

Example: Self-repairing shortest-path

Security and Survivability

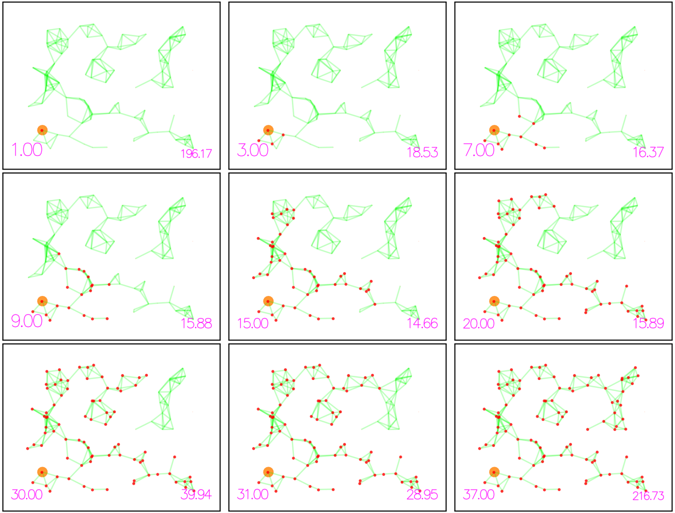
```
(def shortest-path (source destination)
 (letfed
         ;; di is the total distance from source and dest to executing node
         ((di (+ (distance-to source)
                 (distance-to destination)))
         ;; min-di is the shortest path distance
          (min-di (min-hood (broadcast destination di))))
  ;; if executing node is on the shortest path (and not infinity)
  (if (and (not (= min-di (inf)))
           (= min-di di))
   (blue 1)        ;; turn on blue LED
   (blue 0))))     ;; else, turn off blue LED
```

Messaging



```
(mov                    ;; move the device
   (normalize           ;; normalize the vector
      (int-hood         ;; integrate over each neighbor's vector
         (nbr-vec))))   ;; return distance-vector to each neighbor
```

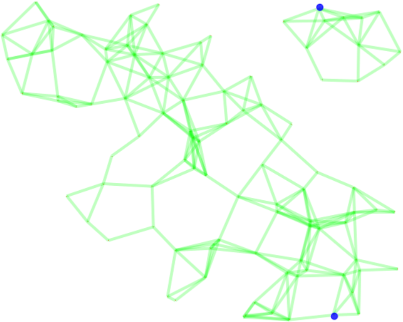
Mobility



Back to Restriction

Conflict Management

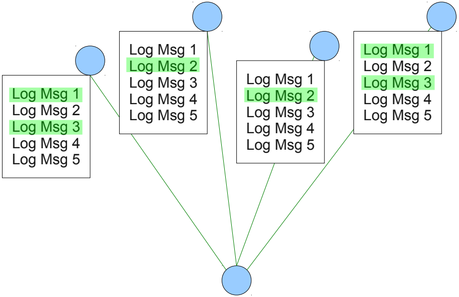
“Facilitates and enables the management of interdependencies between agents activities and decisions.”



Example: the `elect` operator is a self-stabilizing symmetry-breaking function that selects leaders in a cooperative society.

Logging

"Enables information about events that occur during agent system execution to be retained for subsequent inspection."



Log Msg 1
Log Msg 2
Log Msg 3
Log Msg 4
Log Msg 5

Log Msg 1
Log Msg 2
Log Msg 3
Log Msg 4
Log Msg 5

Log Msg 1
Log Msg 2
Log Msg 3
Log Msg 4
Log Msg 5

Log Msg 1
Log Msg 2
Log Msg 3
Log Msg 4
Log Msg 5

Future work for Proto includes implementing logging.
Idea: similar to queries on distributed DB.