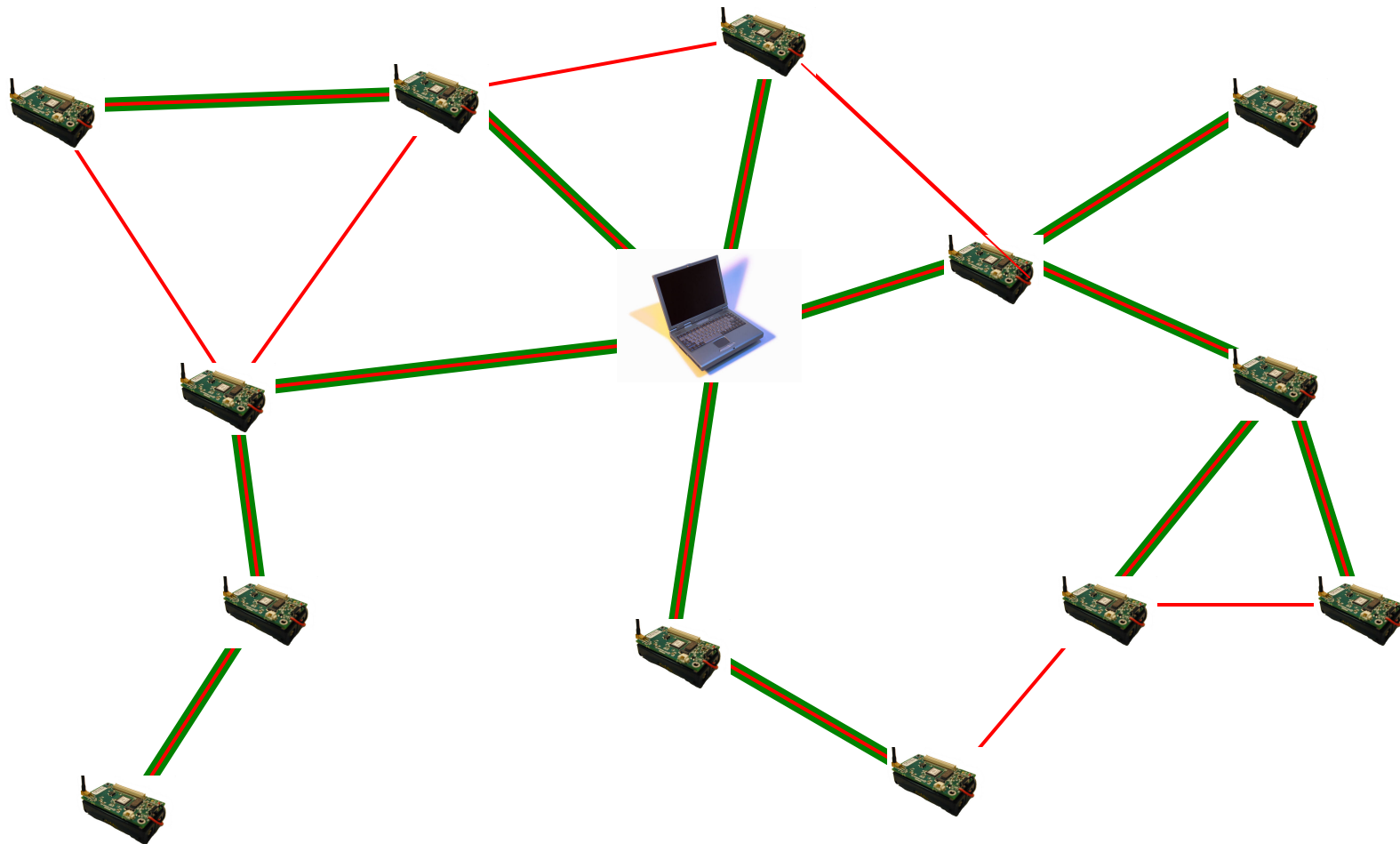


Mobile Agent Applications in Sensor Networks

Chien-Liang Fok
Gruia-Catalin Roman
Mobile Computing Lab
Washington University in St. Louis

[Wireless Sensor Networks]



[Mobile Agents]

**Mobile Agent = Mobile Code + State
+ Intelligence**

- Not to be confused with:
 - Remote evaluation
 - Code on demand
- Two types of state mobility:
 - Strong: migrate data + thread state
 - Weak: migrate data state only

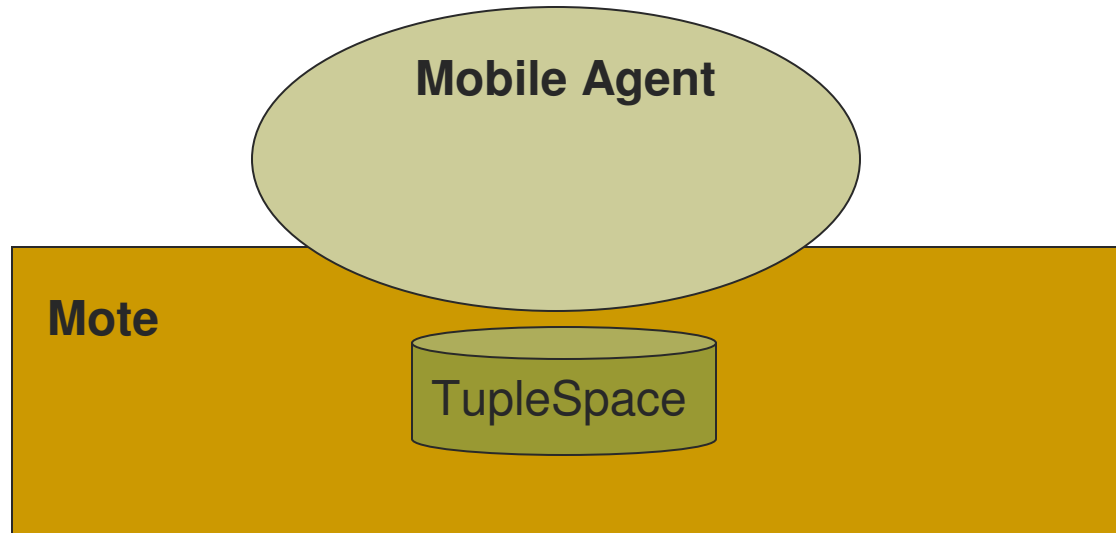
[Why Mobile Agents in S.N.?]

- Generic network
 - Reprogram a network after deployment
- Asynchronous operation
 - Agents operate independently of the creator
- Memory usage
 - Same memory can be used by multiple applications (at different times)
- Active networks
 - Intelligent migration of code and state based on the situation

[Mobile Agent Middleware]

- Mobile agent support should be implemented in **middleware**
- What features should the mobile agent middleware provide?
 - Operation primitives
 - Data and instruction stores (?)
- How can the middleware simplify application development?

[Primitive Operations]



Agent Lifecycle

Inject(Program, Parameters, Destination)
Clone(Parameters, Destination)
Move(Destination)
Expire()

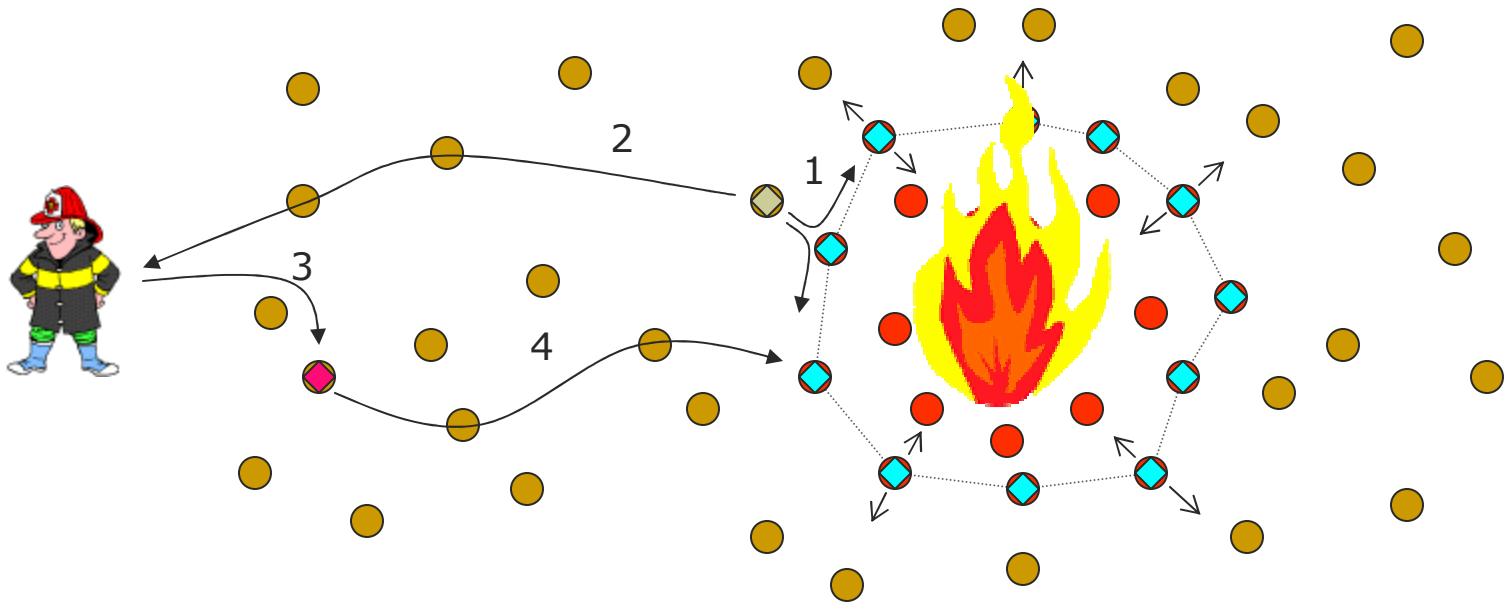
Agent / Data Query

findAgents(Region, Attributes)
findNodes(Region)
findData(Region, Attributes)
grabData(Region, Attributes)

Agent Communication / Data Ops.

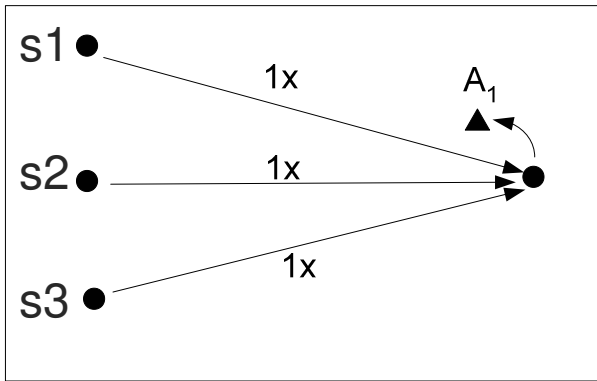
Send(Destination, Data, TimeToLive)
Send(Agent, Data, TimeToLive)

[Fire Detection]



Active Data Aggregation

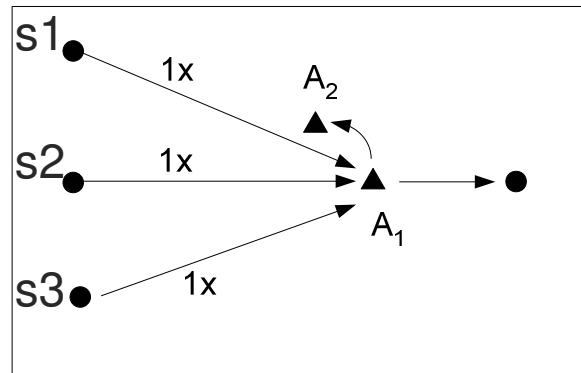
Sensor Network Area



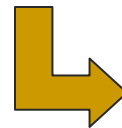
(a)



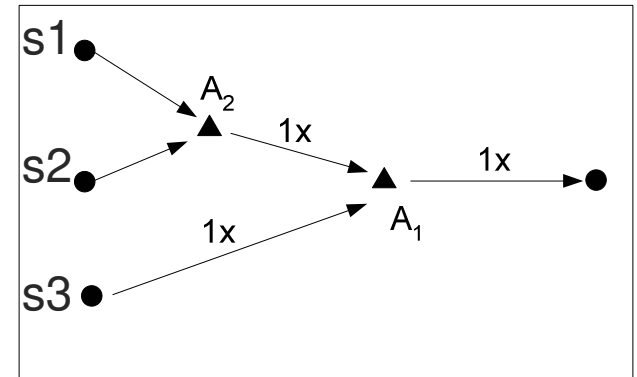
Sensor Network Area



(b)

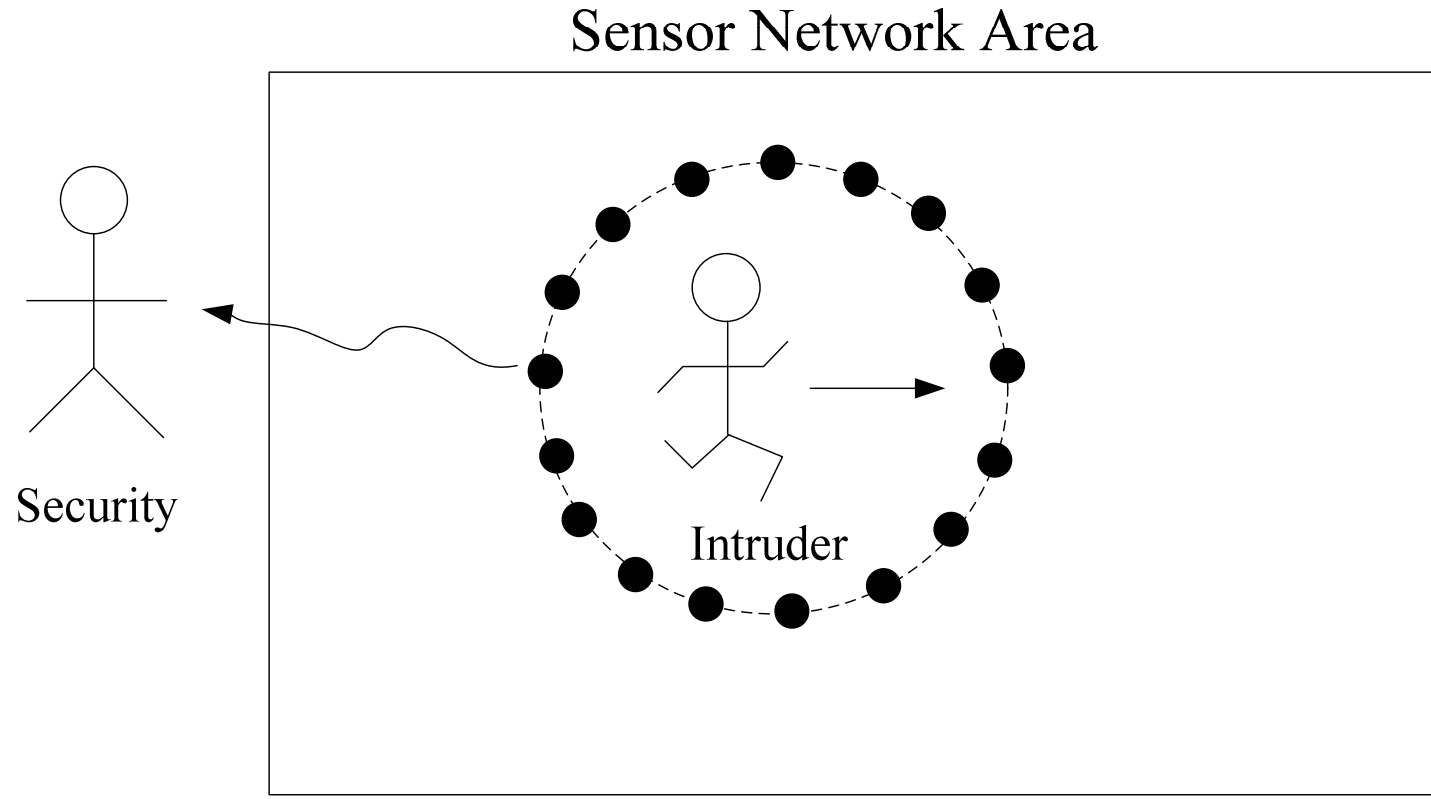


Sensor Network Area



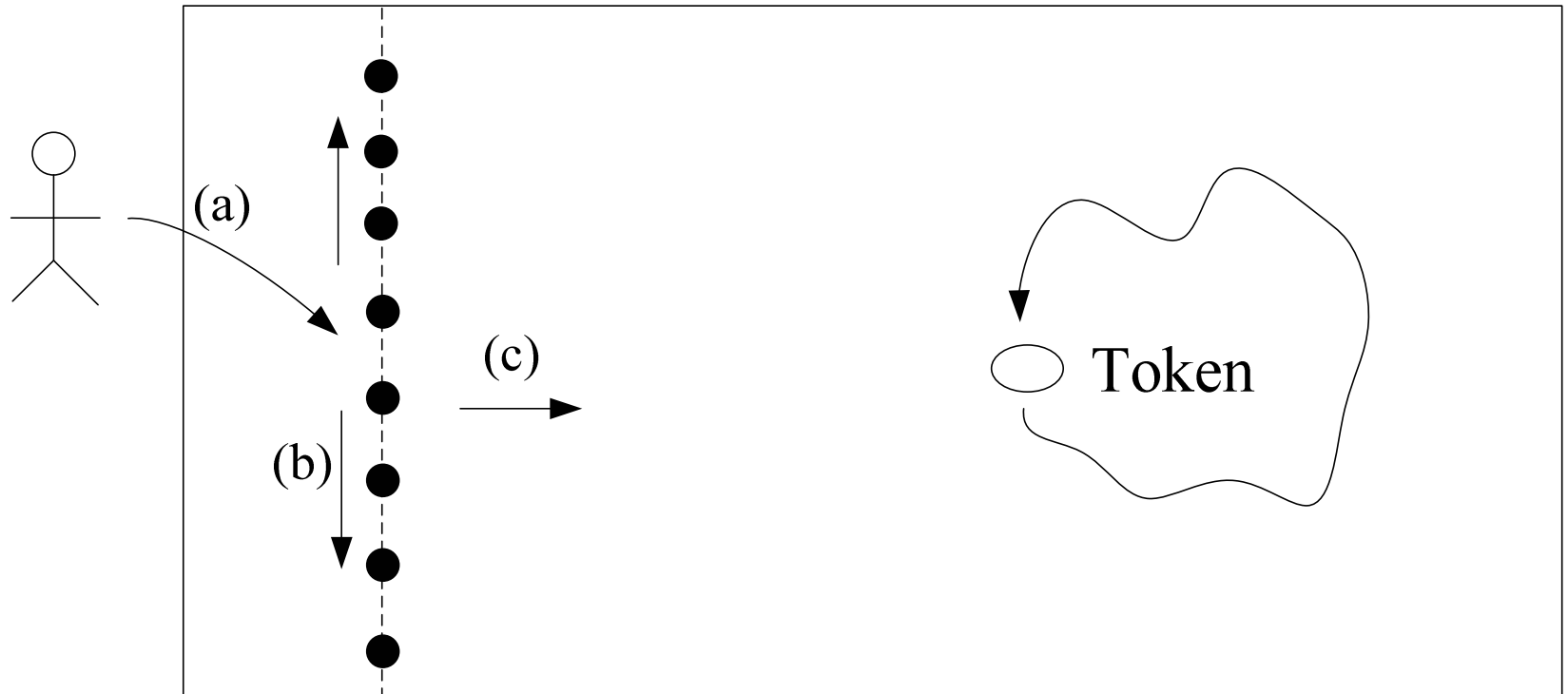
(c)

[Active Tracking]

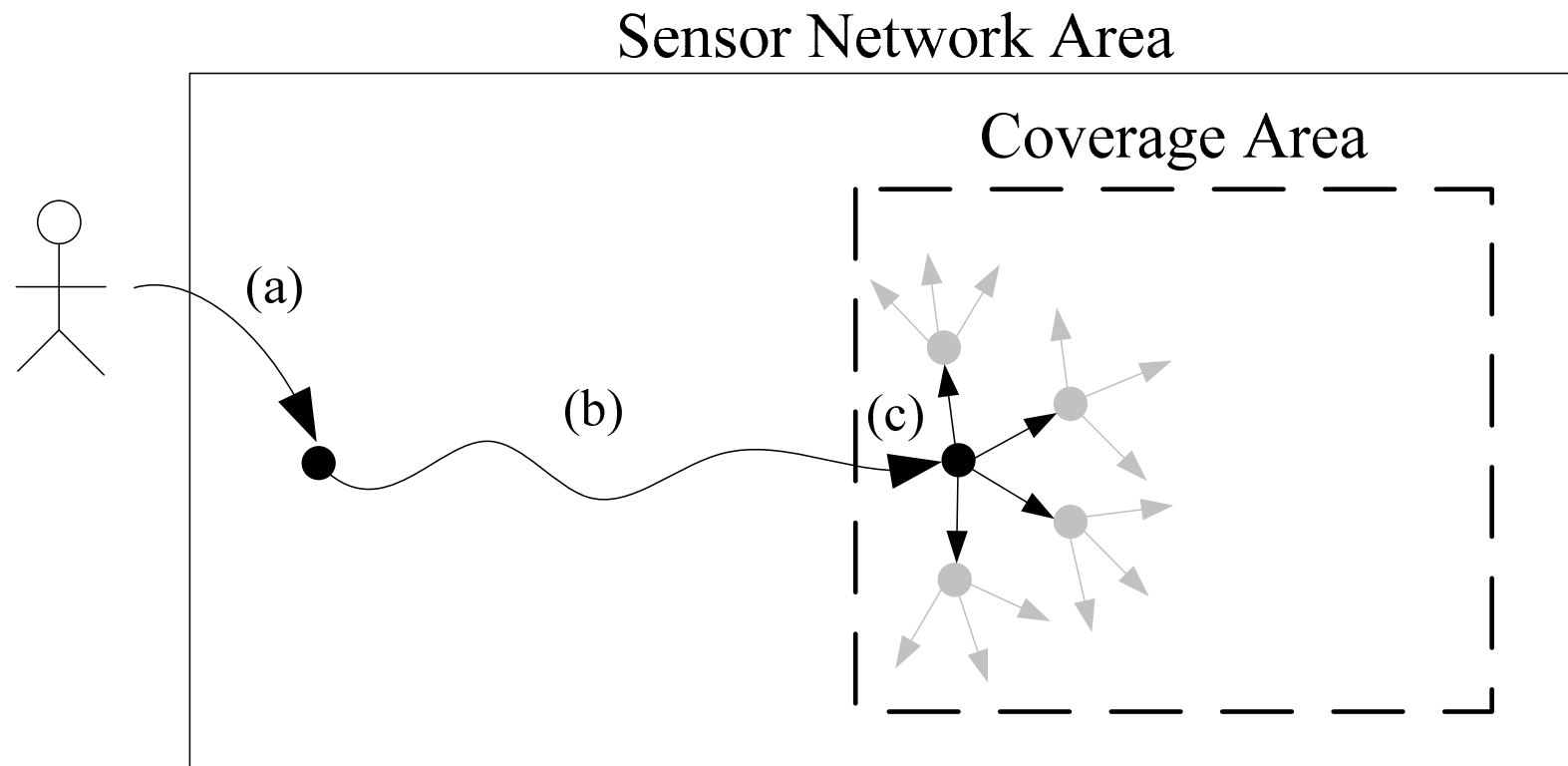


[Sweep-Line Search]

Sensor Network Area



[Sensing Coverage]



[Research Directions]

- Refinement of primitives
 - Are they sufficiently expressive and powerful?
 - Need a formal specification of their semantics
- Implementation
 - Integration within a virtual machine (prototype)
- Evaluation of mobile agent benefits
- Agent evolution
 - Split each agent's code into fragments, mix up the fragments to create new mutant agents