Many-to-Many Invocation and SpatialViews

Justin Collins

CS 239

(Spring 2007)

Problems

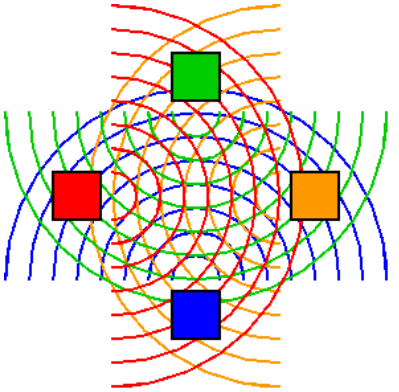
- Mobile ad-hoc networks have special characteristics:
  - Channel quality variations
  - Devices join and leave network frequently
  - No central server
  - Power and resource constraints
- Difficult to write applications using traditional approaches

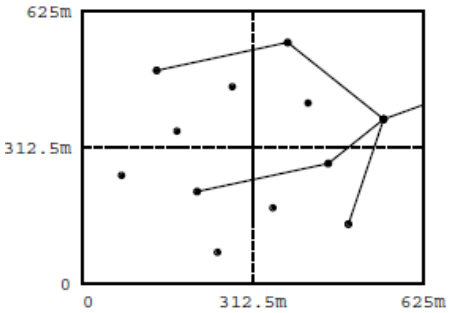
Desired Properties

- Provide general-purpose programming abstractions for mobile ad hoc networks
- Handle frequent disconnections gracefully
- Avoid needing to directly address devices
- Scalable to many devices
- Low power consumption
- Decentralized
- Time efficient, low overhead
- Adjust to channel quality and device capabilities

Many-to-Many Invocation and SpatialViews

- M2MI
  - Broadcast method invocations
  - Object handles
  - Written in Java
- SpatialViews
  - Object iterators
  - Code migration
  - Extends Java

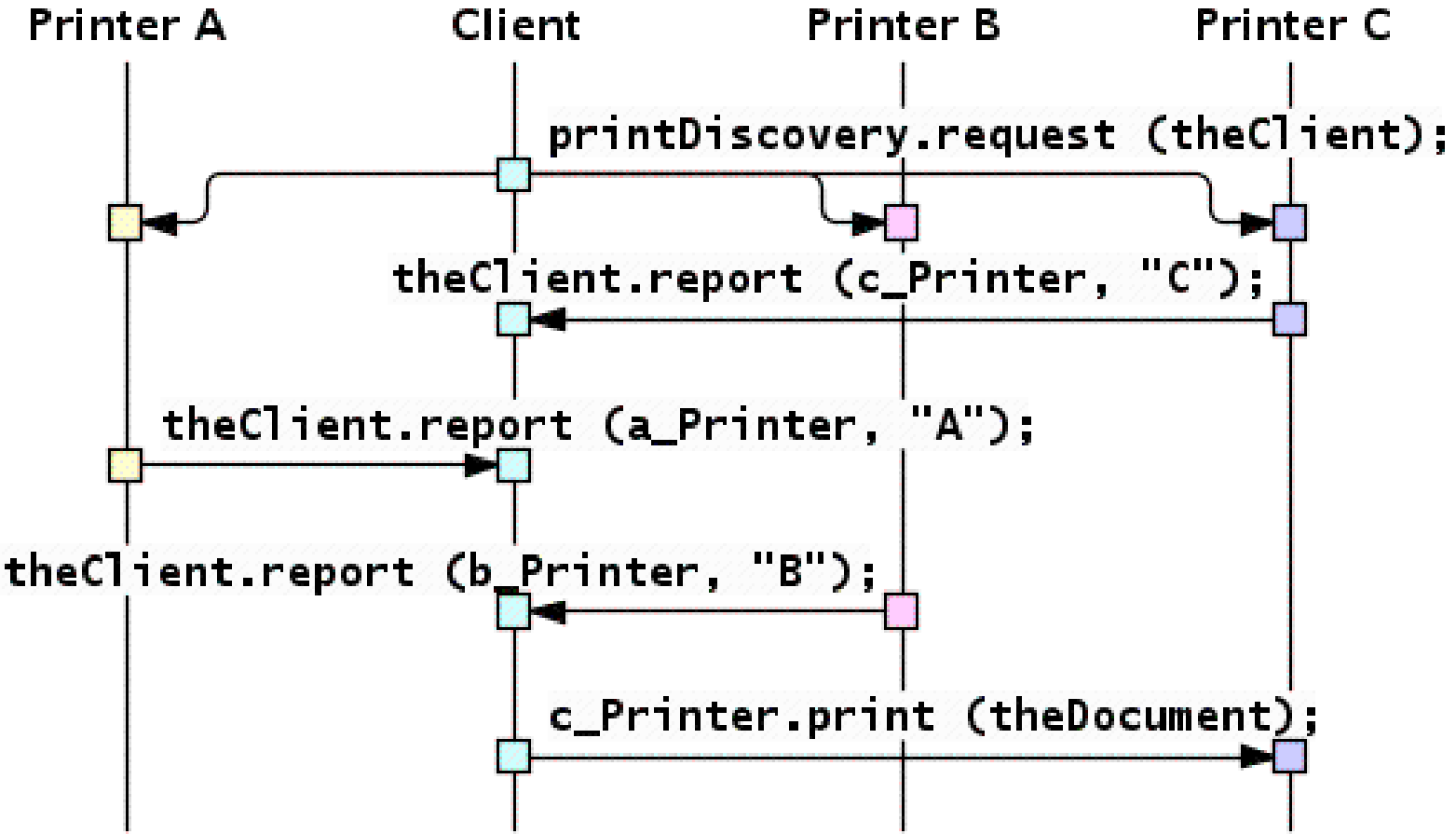




M2MI
Method Calls

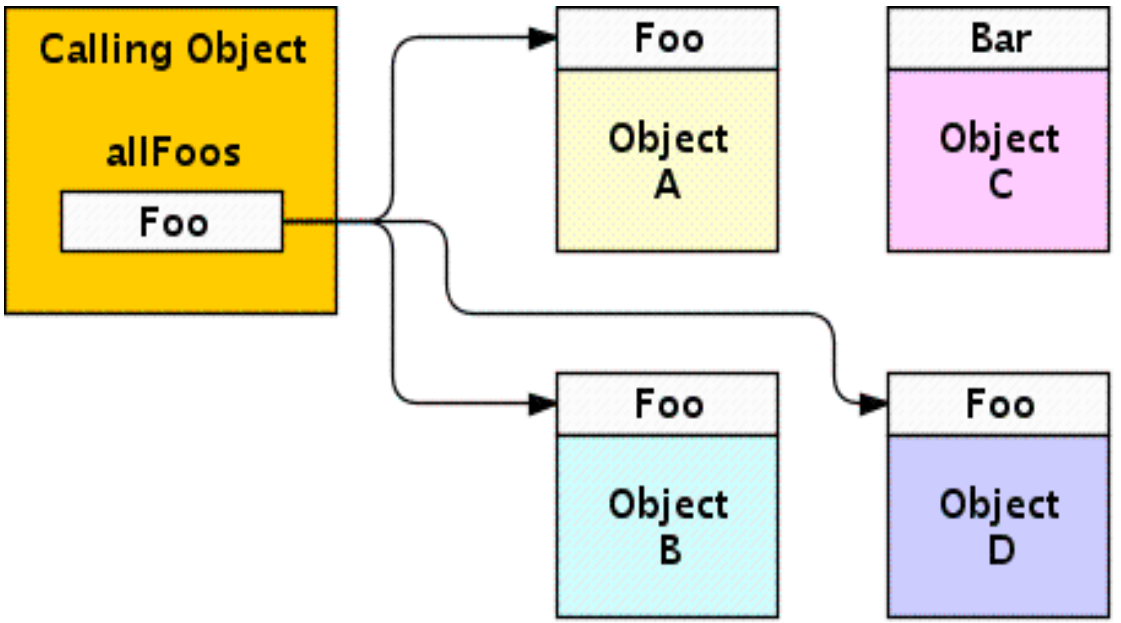
- M2MI uses the inherent broadcast nature of wireless communication
- Method invocations and parameters are broadcast to all nearby nodes
- Method calls to remote objects are actually just message passing
- No return values or shared variables
- Method calls are nonblocking

M2MI
Printer Discovery



From: Kaminsky & Bischof -- Many-to-Many Invocation OOPSLA 2002 -- 07-Nov-2002

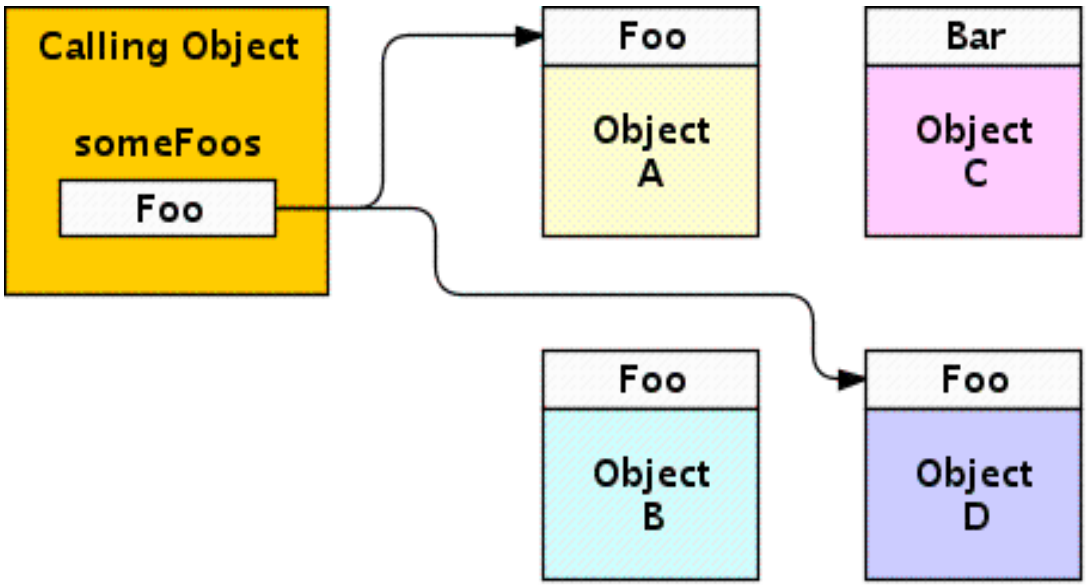
M2MI
Omnihandle




```
Foo allFoos = (Foo) M2MI.getOmnihandle (Foo.class);
```

From: Kaminsky & Bischof -- Many-to-Many Invocation OOPSLA 2002 -- 07-Nov-2002

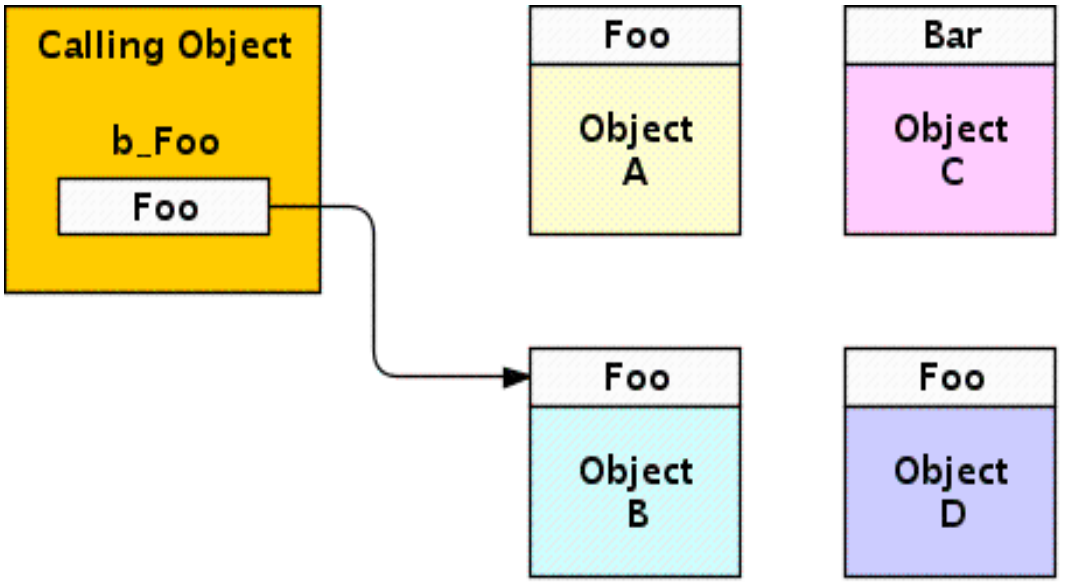
M2MI
Multihandle




```
Foo someFoos = (Foo) M2MI.getMultihandle (Foo.class);
someFoos.attach (a);
someFoos.attach (d);
```

From: Kaminsky & Bischof -- Many-to-Many Invocation OOPSLA 2002 -- 07-Nov-2002

M2MI
Unihandle




```
Foo b_Foo = (Foo) M2MI.getUnihandle (b, Foo.class);
```

From: Kaminsky & Bischof -- Many-to-Many Invocation OOPSLA 2002 -- 07-Nov-2002

M2MI Objects

- Remotely accessible objects are exported to M2MI layer
- Objects can be exported as more than one type
- Exporting objects is done at runtime
- Messages for objects not hosted by a particular device are ignored

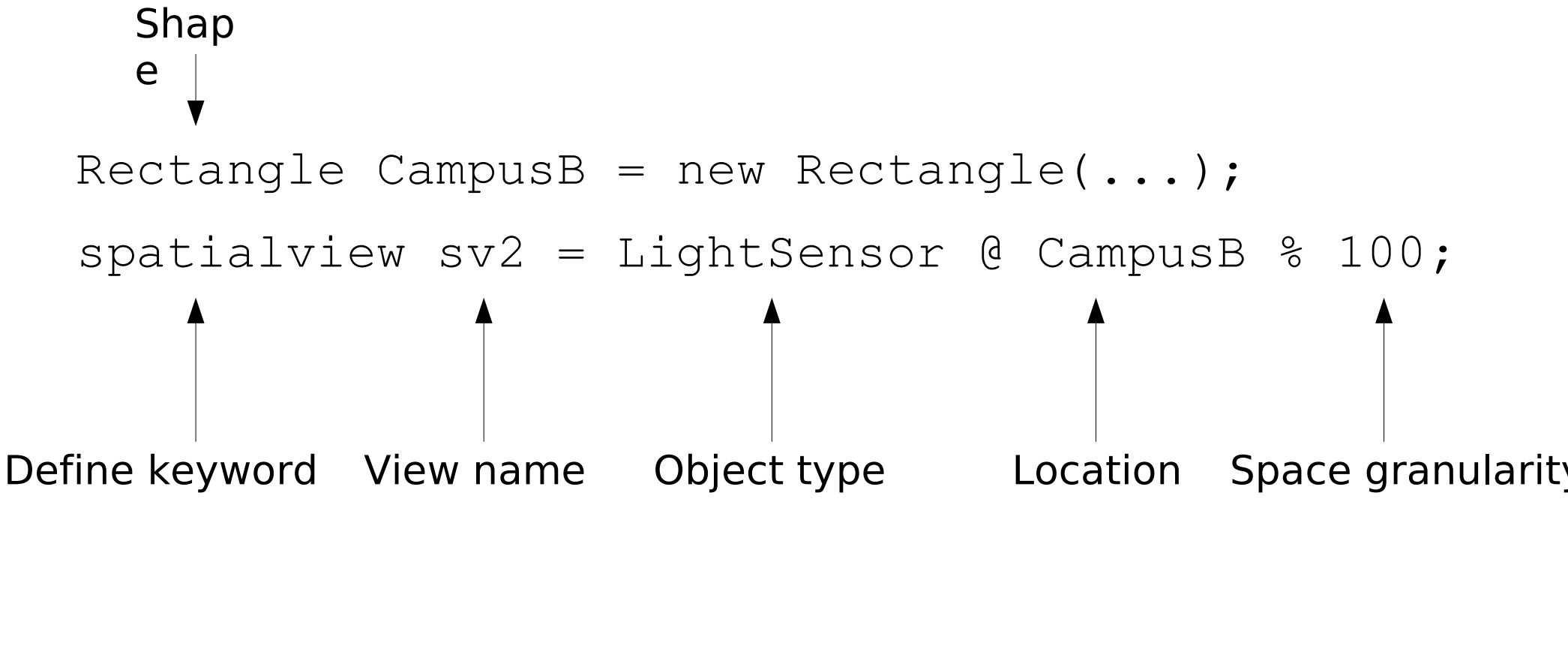
SpatialViews Overview

- Use iterators and code migration to execute code at nodes in the network
- Iterators usually use physical locality
- Variables can be shared within iterators
- A spatial view is a predefined collection of nodes in a given space

SpatialViews

- A spatial view has:
  - type
  - shape
  - location
  - space granularity

SpatialViews
Spatial View



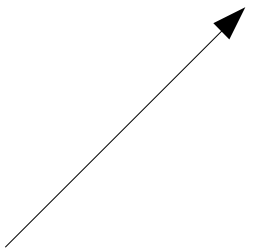
SpatialViews
Iterators

- Iterators visit unique virtual nodes
- A single “real” node can be visited more than once if it moves or a specified amount of time elapses
- Iteration can be infinite or timeout
- Iteration can be simple or use geographic information

SpatialViews Interator

```
spatialview sv2 = LightSensor @ CampusB % 100;


visiteach x : sv2 every 5 within 600 {
   //do stuff
}
```



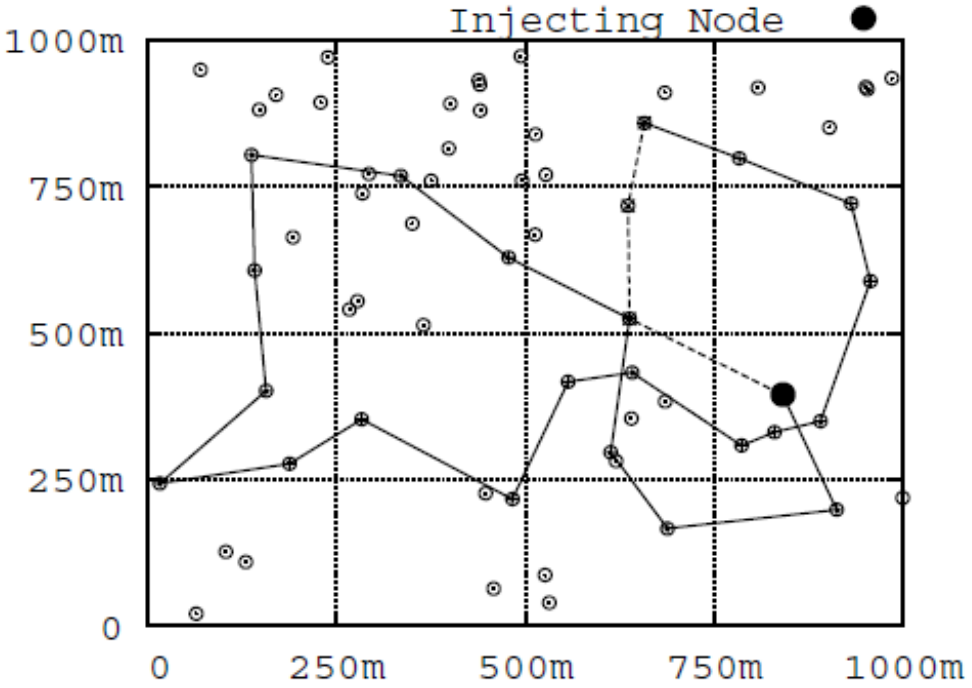
Time difference (seconds)

Overall timeout (seconds)

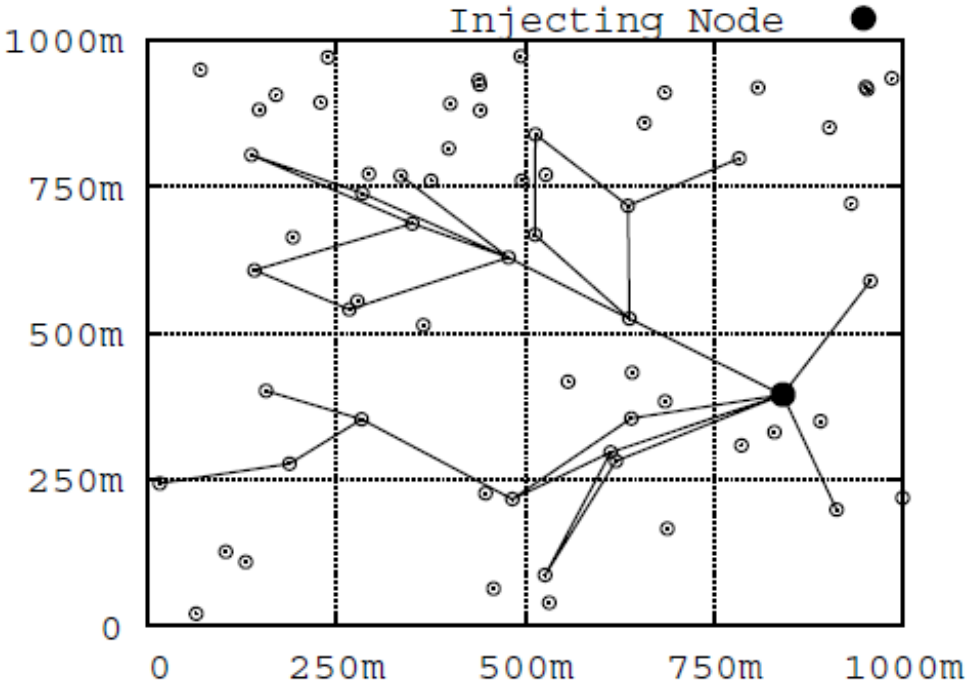
SpatialViews Reduction Example

```
public class AverageLighting {
   public void static main(String[] args) {
      sumreduction float s=0;
      sumreduction int n=0;
      spatialview sv=LightSensor @ CampusB % 250;
      visiteach x : sv {
         s += x.read(); n++;
      }
      if (n>0)
         System.out.println(Float.toString(s/n));
   }
}
```

SpatialViews Reduction Example



Geographic Serial Iteration



Geographic Flooding

M2MI Advantages

- Decentralized
- No addressing or discovery schemes
- Intended for low-power devices
- Devices can come and go without a problem
- Simple semantics

SpatialViews Advantages

- High-level constructs
- Able to spread computation over devices
- Shared variables
- Efficiency through geographic iteration

M2MI Drawbacks

- No return values for method invocations
- Method invocations are basically just message passing
- Relies on the broadcast range of devices
- No forwarding or routing of messages
- Lack of higher-level constructs

SpatialViews Drawbacks

- Devices need locality information
- Devices assumed to have substantial computation power
- Hard to directly address a single device
- 'Injecting' node could be considered a centralized source. If data needs to be reduced and propagated back to the initial node and the initial node disappears, the computation breaks down

Comparison

	M2MI	SpatialViews	iMASH
General Purpose			
Disconnection Handling			
Indirect Addressing			
Scalability			
Low Power			
Decentralized			
Time efficient			
Channel quality aware			
Device capability aware			

References

- *Programming for Ad-Hoc Networks of Mobile and Resource Constrained Devices* - Yang Ni, Ulrich Kremer, Adrian Stere and Liviu Iftode, ACM SIGPLAN 2005 Conference on Programming Language Design and Implementation (PLDI), Chicago, Illinois, June 2005.
- *Many-to-Many Invocation: A new framework for building collaborative applications in ad hoc networks.* Hans-Peter Bischof and Alan Kaminsky. CSCW 2002 Workshop on Ad Hoc Communication and Collaboration in Ubiquitous Computing Environments, New Orleans, Louisiana, USA, November 2002.
- *iMASH: Interactive Mobile Application Session Handoff* R. Bagrodia, S. Bhattacharyya, F. Cheng, S. Gerding, G. Glazer, R. Guy, Z. Ji, J. Lin, T. Phan, E. Skow, M. Varshney, and G. Zorpas. In Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services (Mobisys 2003), May 2003.