Edinburgh Napier UNIVERSITY

An Application of CoSMoS Design Methods to Pedestrian Simulation

Sarah Clayton
Dr Neil Urquhart
Prof Jon Kerridge

Edinburgh Napier UNIVERSITY

Introduction

- Pedestrian modelling
- Experiences with using JCSP
- Employing the CoSMoS design method
  (... and getting it wrong)
- Simulation

Pedestrian modelling

- ... is difficult
- Is done at many levels of scale
- Can be (but usually isn't) data driven
- Our approach to the problem is:
  - At a relatively small scale
  - Data driven

Edinburgh Napier UNIVERSITY

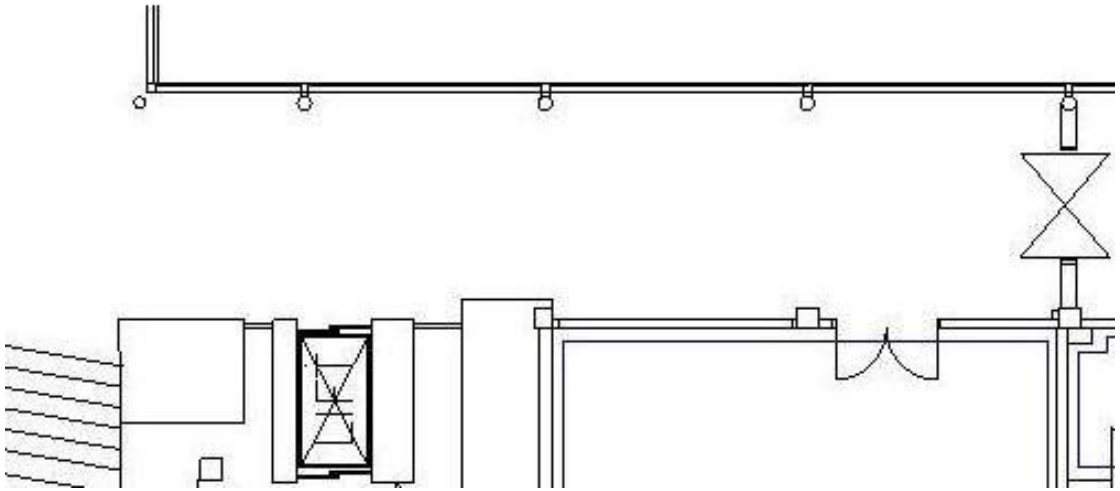
Experiences of using JCSP

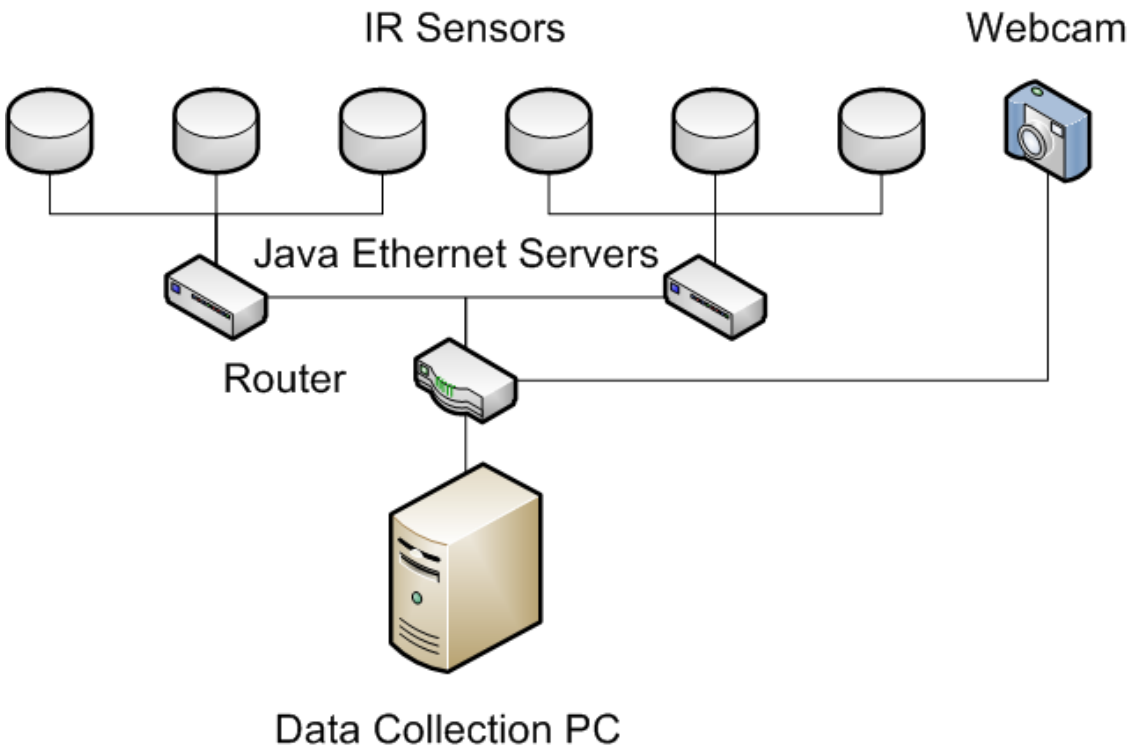
- Watching Walkers exhibit at 2006 Edinburgh Science Festival
  - 5000 visitors to exhibit
  - System performed incredibly robustly
  - LOC < 3000
- Data collection
  - Six infrared detectors deployed in corridor
  - Read concurrently in real-time
  - Unattended, automated operation
  - More data than we know what to do with
- Simulation
  - Robust framework to build on
  - Deadlock, livelock, race hazard free
  - No thread programming

Edinburgh Napier UNIVERSITY

Experimental Area

Experimental Area




IR Sensors
Webcam
Java Ethernet Servers
Router
Data Collection PC

CoSMoS design method

- What they are
- How I got it wrong
- Good results nonetheless
- Matters of scale

Edinburgh Napier UNIVERSITY

CoSMoS design method

- Built around phased synchronisation on barriers
- Combines this with the use of client server architecture
- Both proven to be error free, if used correctly

Edinburgh Napier UNIVERSITY

Example Code - Barriers

```
public class Agent implements CSProcess {
    private Barrier discover;
    private Barrier modify;

    public Agent {
         discover.enroll();
         modify.enroll();
    }

    public void run() {
         while(true) {
                  discover.sync();
                  discover();
                  modify.sync();
                  modify();
         }
    }
}
```

Example Code – Client/Server

Edinburgh Napier UNIVERSITY

Client

```
private ChannelOutput request;
private AltingChannelInput response;

...

request.write(Request.DISCOVER);


Info info = (Info) response.read();

...
```

Server

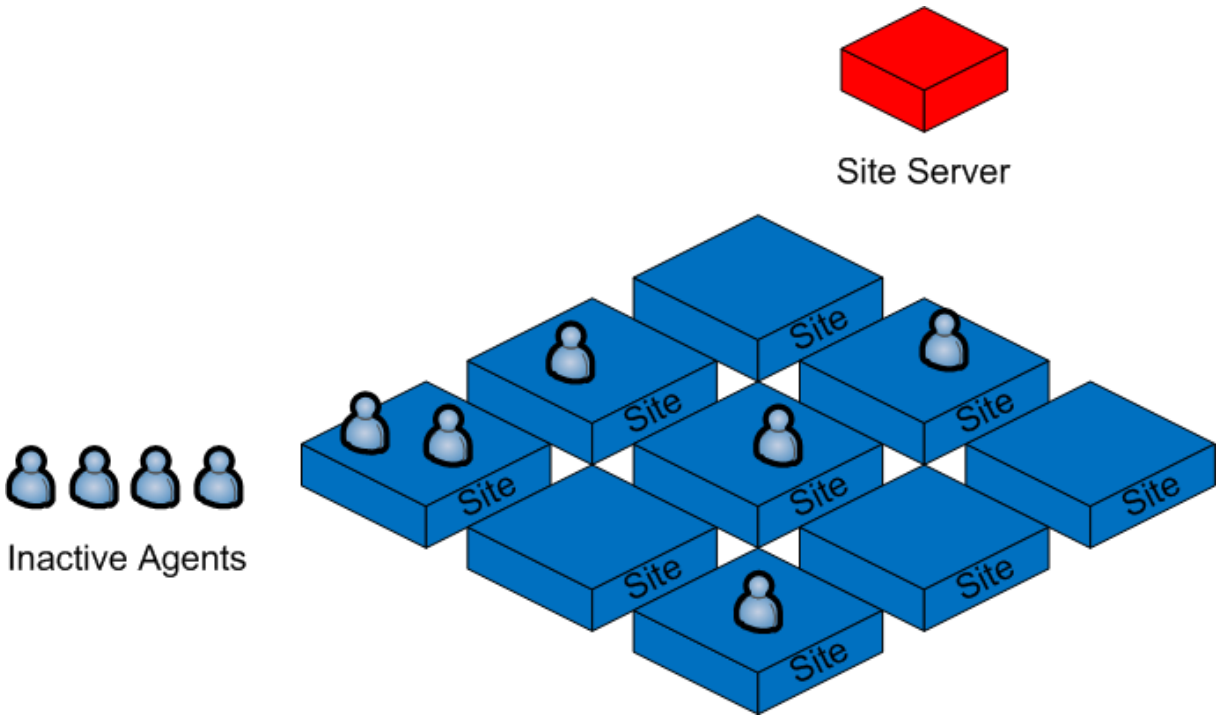
```
private AltingChannelInput request;
private ChannelOutput response;

...

Request r = (Request)request.read();
switch(r) {
   case Request.DISCOVER:
        response.write(discover());
        break;
...
}
```

Edinburgh Napier UNIVERSITY

How I got it wrong



Edinburgh Napier UNIVERSITY

How I got it wrong

Agent		Site		Site Server
		Synchronise on discover barrier		
		Request global coordinates	→	Receive requests
		Receive global coordinates	←	Send global coordinates
Request update	→	Receive requests		
Receive update	←	Send global coordinates		
		Synchronise on modify barrier		
Modify state				
Send state	→	Receive state		
Receive ACK	←	Send ACK		
		Send updates	→	Receive updates
		Receive ACK	←	Send ACK
				Aggregate updates into global coordinates

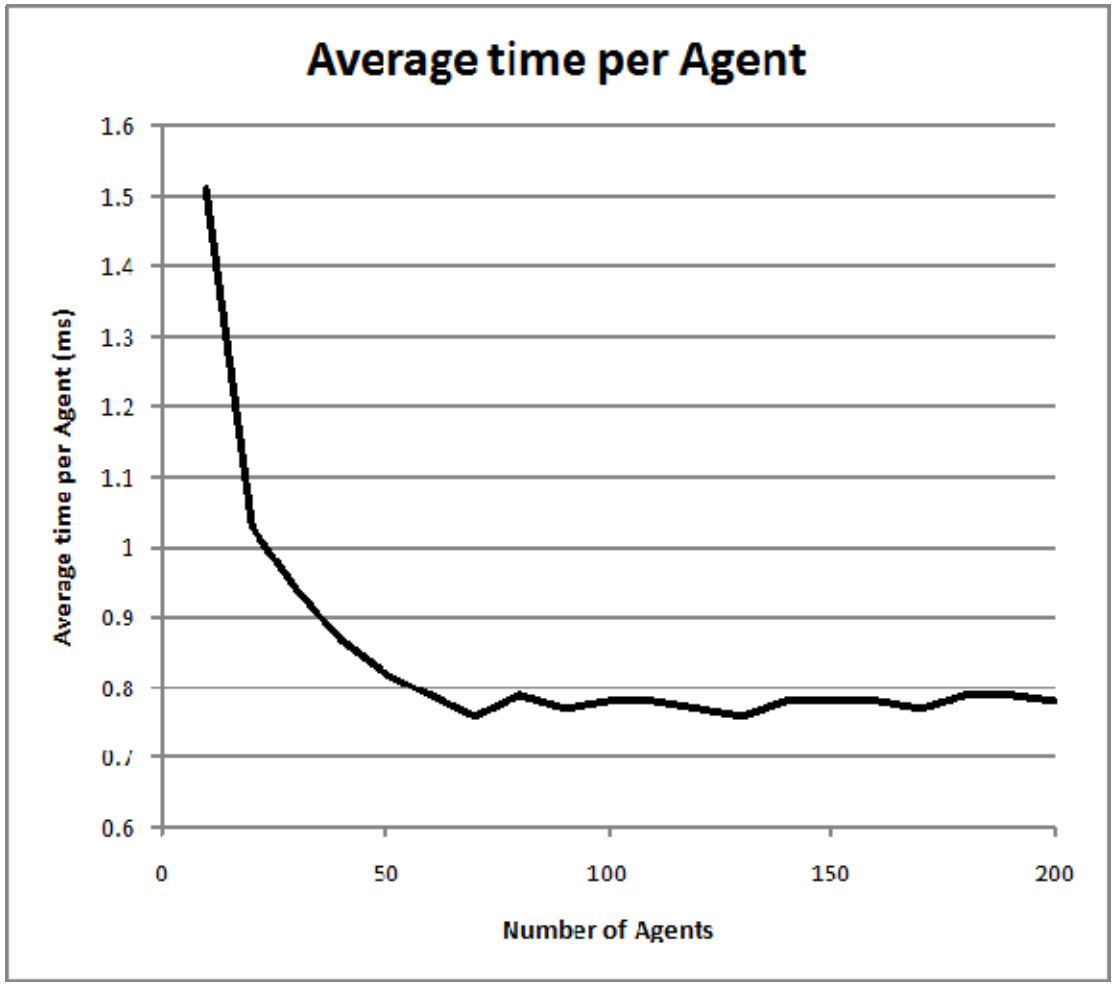
Edinburgh Napier UNIVERSITY

Processes that express space are functionally and conceptually separate from time contingent processes

(i.e. Don’t engage in barrier synchronisation)

Edinburgh Napier UNIVERSITY

Good results nonetheless



Matters of scale

Edinburgh Napier UNIVERSITY

CoSMoS

- Millions of processes
- Macroscopic scale
- Built on occam
- Low process overhead

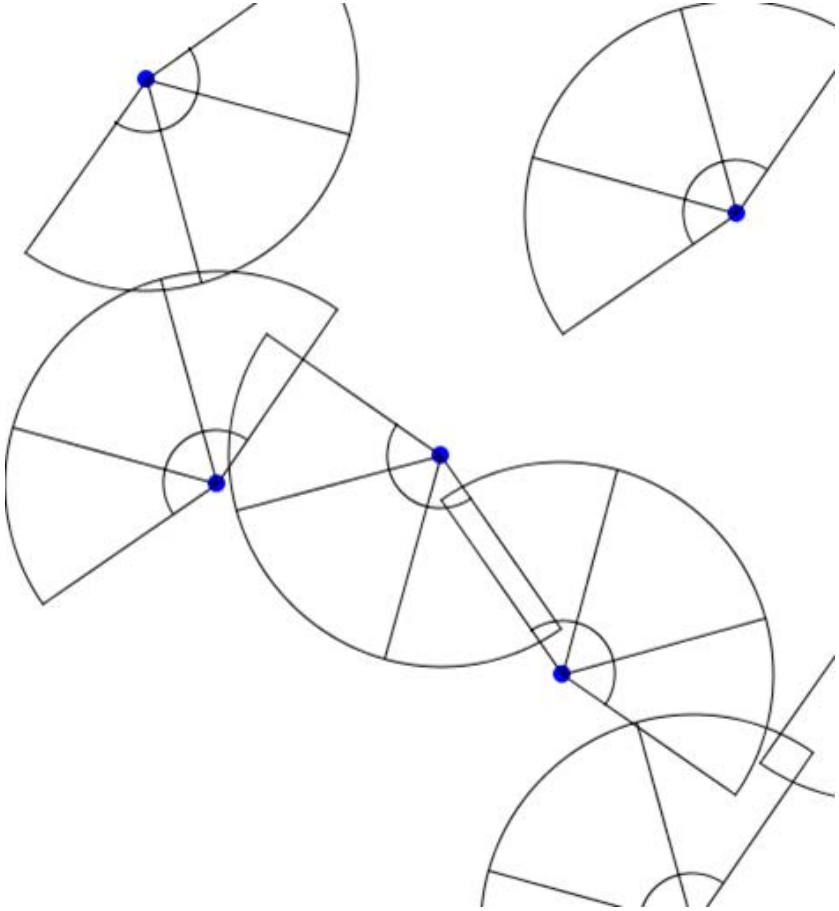
Pedestrian Model

- Tens of processes
- Microscopic scale
- Built on JCSP
- High process overhead

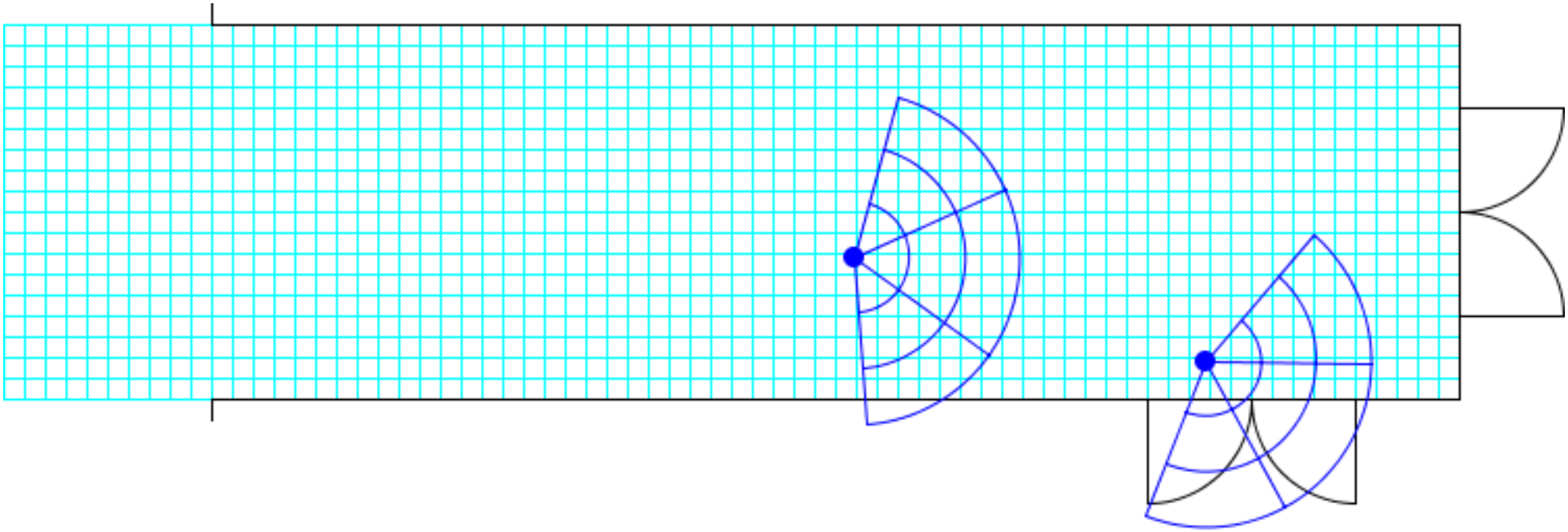
Edinburgh Napier UNIVERSITY

Examples

Edinburgh Napier
UNIVERSITY



Edinburgh Napier
UNIVERSITY



Edinburgh Napier UNIVERSITY

Conclusion

- Utility of JCSP and CoSMoS
  - Significantly reduced development time
  - Less error prone
  - More functionality for less code

Edinburgh Napier UNIVERSITY

Future Work

- Building a Learning Classifier System to extract behavioural parameters from the observed trajectories.
- Particular focus on collision avoidance.
- Building a model to enable analysis of built environment, based on real, observed parameters.

Edinburgh Napier UNIVERSITY

Thank You