Using Stigmergy to Co-ordinate Pervasive Computing Environments

Peter Barron and Vinny Cahill,
Distributed Systems Group,
Department of Computer Science,
Trinity College,
Dublin 2, Ireland.

Email: {Peter.Barron, Vinny.Cahill}@cs.tcd.ie

6th IEEE Workshop on Mobile Computing Systems & Applications

Introduction

- Pervasive computing environments
  - Computation everywhere.
  - Reacting intelligently though in a manner that is unobtrusive.
- Possible to convert many everyday spaces into pervasive computing environments.
  - Meeting rooms, office spaces, lecture theaters.
- ***Ad-hoc gathering*** of autonomous entities.
  - Example: urban setting.
- ***Objective of the work*** presented here is to provide a framework supporting the development of these types of pervasive environments.

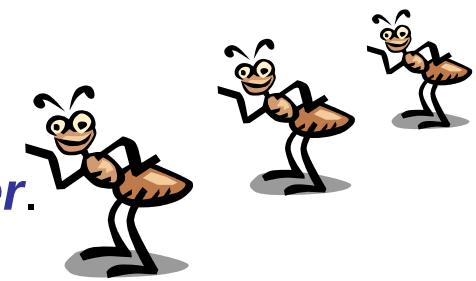
The Challenge…

- Need to be able to ***coordinate*** collections of ***interacting*** entities.
  - No central authority for coordinating entity behavior.
  - Require a highly ***decentralized method of organizing*** components.
  - Pervasive computing environments evolve in a particular area.
- Have to handle the highly ***dynamic*** and ***unpredictable*** nature of these type of environments.
  - Ability to adapt to changing configurations.
  - The increased mobility of entities.
- Need to allow the ***incremental*** construction and ***improvement*** of solutions without adversely effecting the rest of the environment.
  - Ad-hoc composition of pervasive computing environments.
  - Not possible to reboot the environment, to large.
- ***Spontaneous interaction*** and ***physical integration*** of entities into the environment.

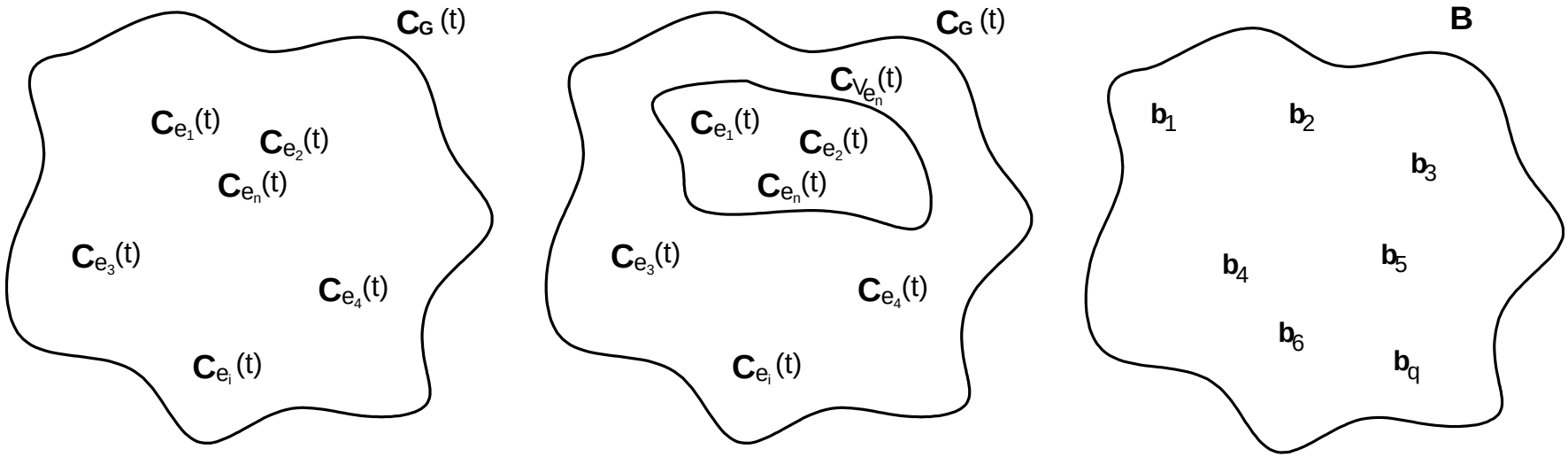
A Biologically Inspired Approach

- Stigmergy (Grassé 1959)
  - how ***social insects coordinate*** their actions.
  - ***Indirect*** communication ***mediated*** through the ***local*** environment.
  - A highly decentralized mechanism of coordinating a system.

- Idea of ***simple insects*** maps well to pervasive computing where small devices are spread across the environment.
- ***Indirect communication*** mechanisms decouple entities.
  - As a result the overall system is less fragile and more stable to disturbances in the environment.
  - The system can grow organically and decay gracefully
- Spontaneous interaction between entities can be achieved as communication is ***mediated*** through the ***common medium*** of the environment.
  - A common interoperation model.
- Provides a means of ***coordinating*** the behavior of a pervasive environment in a ***distributed manner***.



A Stigmergic Model for Pervasive Computing

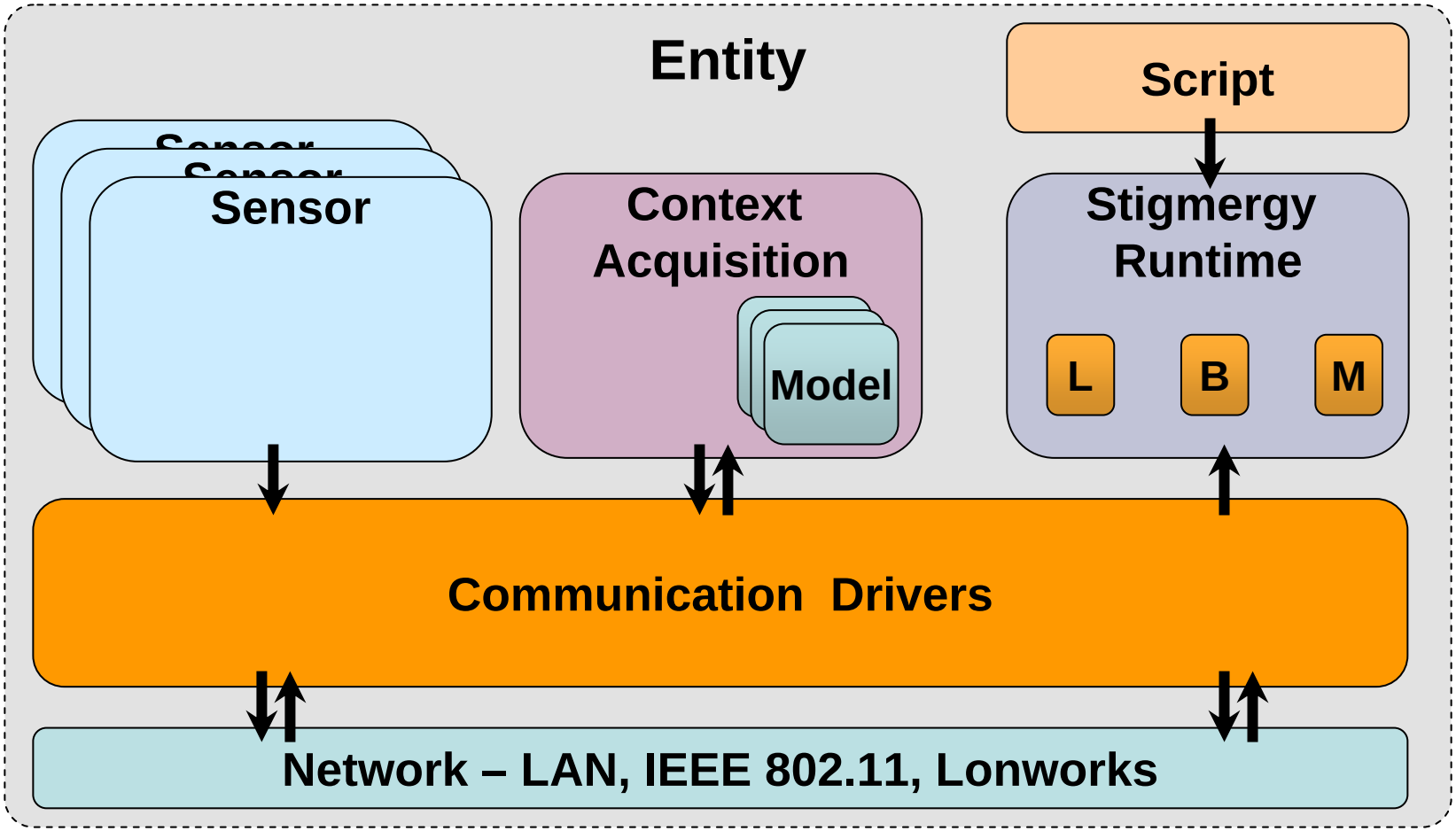


$$\mathbf{C}_{V_{e_n}}(t) = \left\{ \mathbf{C}_{e_i}(t) : \mathbf{C}_{e_i}(t) \in \mathbf{C}_G(t) \wedge \mathbf{L}(e_i, e_n) \right\}$$

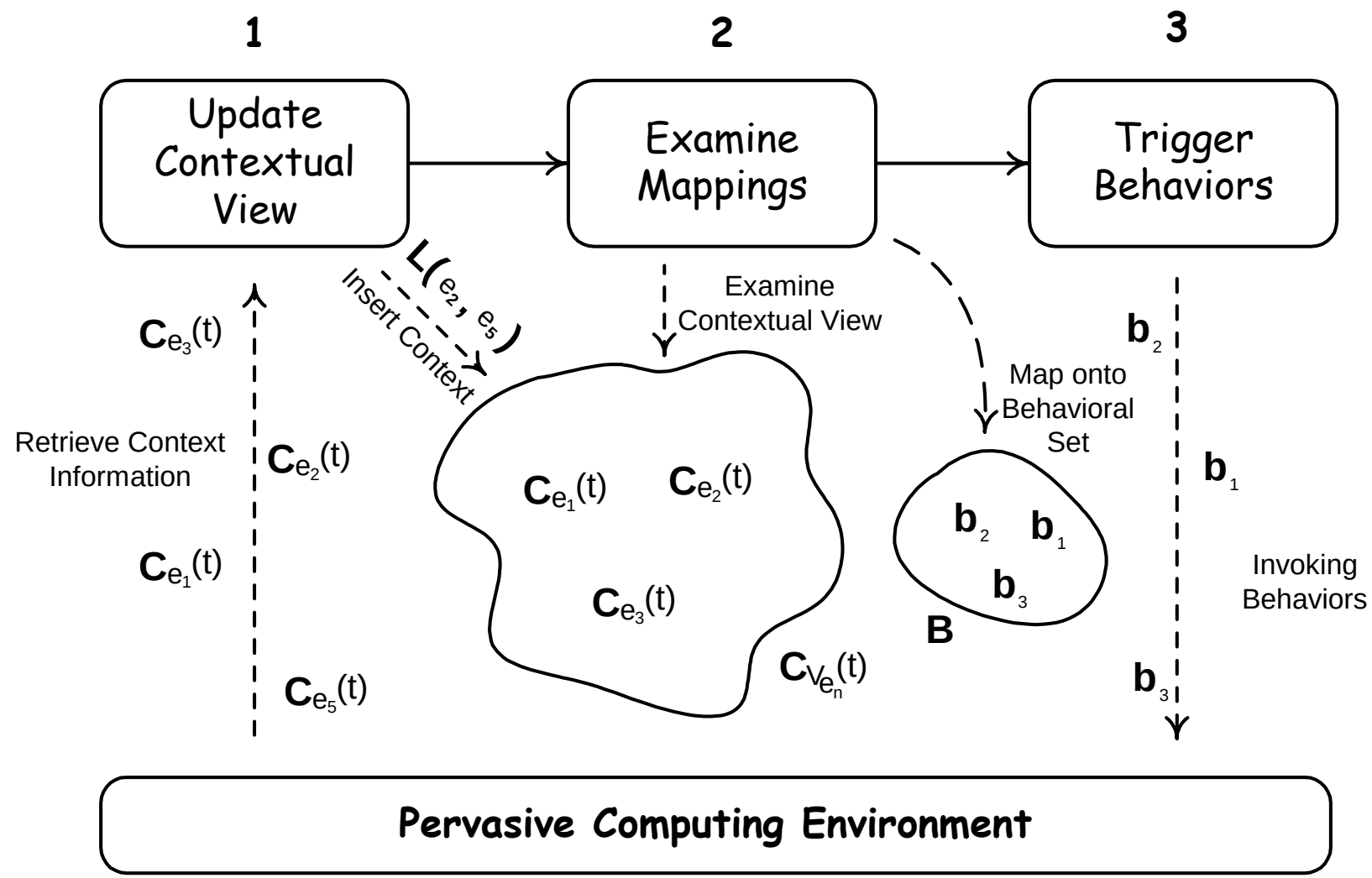
$$\mathbf{M} : \mathbf{C}_{V_e} \rightarrow P(\mathbf{B})$$

The proximity function ***L***, the behavioral set ***B***, and the ***M*** function provide ***three primitives*** used by the framework to define how individual entities respond to changes in the local environment.

Cocoa Architecture Overview



Stigmergy Runtime

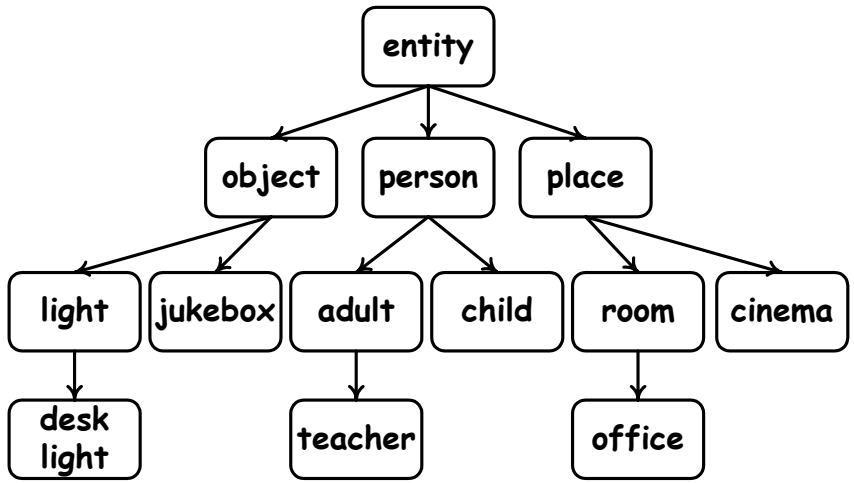


YABS

- ***High-level*** scripting language.
- Used to ***initialize*** the runtime.
- The foundation for the language is built upon the three primitives: ***L***, ***B***, and ***M***
  - Defines what is ***local*** to the entity.
  - Defines the ***behaviors*** that an entity is capable of performing.
  - Provides a method of ***mapping*** an entity's contextual view onto its behavioral set.

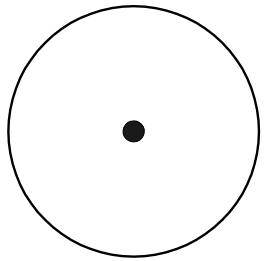
YABS - Example

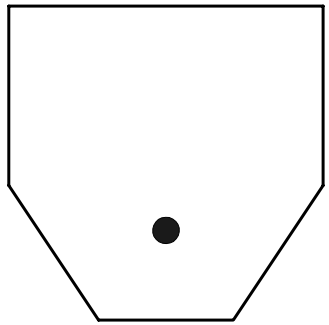
```
jukebox extends object{
        proximity(10)
        behavior play = "ie.tcd.cs.JukeBoxPlayInteraction"
        behavior stop ="ie.tcd.cs.JukeBoxStopInteraction"

        context SomePerson
        SomePerson.person = any
        SomePerson.music = any
        context JukeBoxPlay
        JukeBoxPlay.object = this.object
        JukeBoxPlay.activity = "play"
        context JukeBoxStop
        JukeBoxStop.object = this.object
        JukeBoxStop.activity = "stop“

        map  [JukeBoxStop]  [JukeBoxStop, SomePerson] onto {
                play( majority(SomePerson))
        }

        map [JukeBoxPlay, SomePerson] [JukeBoxPlay] onto{
                stop()
        }
}
```

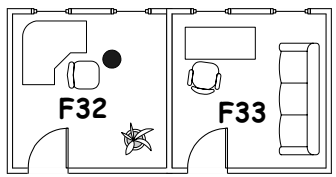


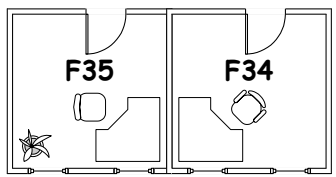
YABS - Example

```
jukebox extends object{
        proximity(10)
        behavior play = "ie.tcd.cs.JukeBoxPlayInteraction"
        behavior stop ="ie.tcd.cs.JukeBoxStopInteraction"

        context SomePerson
        SomePerson.person = any
        SomePerson.music = any
        context JukeBoxPlay
        JukeBoxPlay.object = this.object
        JukeBoxPlay.activity = "play"
        context JukeBoxStop
        JukeBoxStop.object = this.object
        JukeBoxStop.activity = "stop“

        map  [JukeBoxStop]  [JukeBoxStop, SomePerson] onto {
                play( majority(SomePerson))
        }

        map [JukeBoxPlay, SomePerson] [JukeBoxPlay] onto{
                stop()
        }
}
```



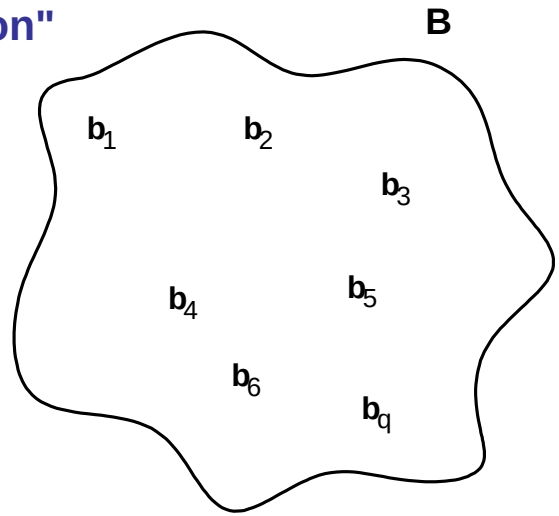






YABS - Example

```
jukebox extends object{
    proximity(10)
    behavior play = "ie.tcd.cs.JukeBoxPlayInteraction"
    behavior stop ="ie.tcd.cs.JukeBoxStopInteraction"

    context SomePerson
    SomePerson.person = any
    SomePerson.music = any
    context JukeBoxPlay
    JukeBoxPlay.object = this.object
    JukeBoxPlay.activity = "play"
    context JukeBoxStop
    JukeBoxStop.object = this.object
    JukeBoxStop.activity = "stop“

    map  [JukeBoxStop]  [JukeBoxStop, SomePerson] onto {
            play( majority(SomePerson))
    }

    map [JukeBoxPlay, SomePerson] [JukeBoxPlay] onto{
            stop()
    }
}
```

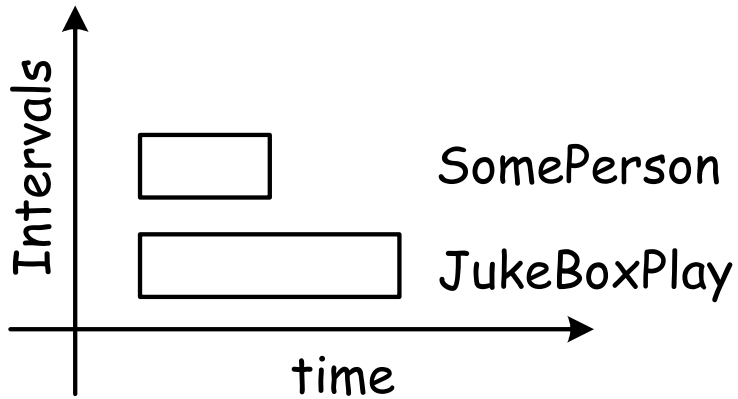


YABS - Example

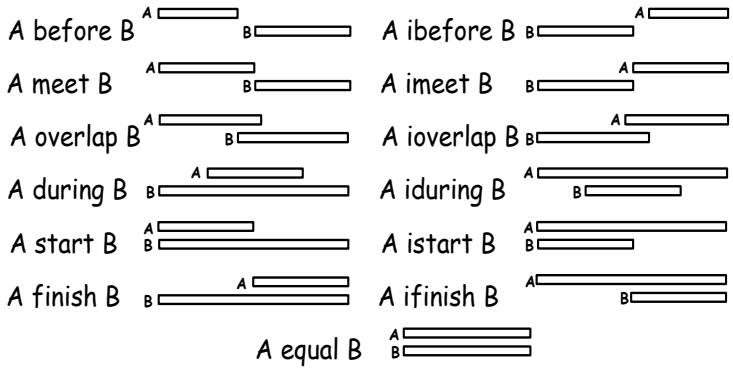
```
jukebox extends object{
        proximity(10)
        behavior play = "ie.tcd.cs.JukeBoxPlayInteraction"
        behavior stop ="ie.tcd.cs.JukeBoxStopInteraction"

        context SomePerson
        SomePerson.person = any
        SomePerson.music = any
        context JukeBoxPlay
        JukeBoxPlay.object = this.object
        JukeBoxPlay.activity = "play"
        context JukeBoxStop
        JukeBoxStop.object = this.object
        JukeBoxStop.activity = "stop“

        map  [JukeBoxStop]  [JukeBoxStop, SomePerson] onto {
                play( majority(SomePerson))
        }

        map [JukeBoxPlay, SomePerson] [JukeBoxPlay] onto{
                stop()
        }
}
```

YABS - Example

```
jukebox extends object{
        proximity(10)
        behavior play = "ie.tcd.cs.JukeBoxPlayInteraction"
        behavior stop ="ie.tcd.cs.JukeBoxStopInteraction"

        context SomePerson
        SomePerson.person = any
        SomePerson.music = any
        context JukeBoxPlay
        JukeBoxPlay.object = this.object
        JukeBoxPlay.activity = "play"
        context JukeBoxStop
        JukeBoxStop.object = this.object
        JukeBoxStop.activity = "stop“

        map  [JukeBoxStop]  [JukeBoxStop, SomePerson]
        onto {
                play( majority(SomePerson))
        }

        map [JukeBoxPlay, SomePerson] [JukeBoxPlay]
        onto{
                stop()
        }
}
```



Allen's Temporal Intervals



Current Implementation

- Developed in ***Java 1.4.2***
  - Core (12000 lines of code, 200 classes)
- Currently running on ***Linux***.
- Communication Driver: ***Steam*** (Meier et al 2002)
  - ***Event-based*** middleware service.
  - Designed for use in ad-hoc networks and with pervasive computing in mind.
  - No ***centralized component***.
  - ***Dynamic subscriptions*** to nearby producers
  - Events can be ***filtered based on the proximity*** of one entity to another.

Evaluation

- Urban Setting
  - Wesland Row
    - Street located in heart of Dublin.
    - 250 meters long
    - Cafes, newsagents, shops, pubs, train stations.
- WAND
  - Ad-hoc network
    - AODV (Perkins et al 1999)
  - Covers approx 2km, 12 nodes.
- Developed and deployed a number of entities using the Cocoa framework.
  - Siopa, Jukebox, Firefox, Punter

Experiences

- Entities are able to ***coordinate*** their behavior.
- Able to construct the environment in an ***incremental*** fashion.
- Able to ***improve solutions*** over time without effecting the rest of the system.
- Fewer dependences between entities appeared to make the overall system ***less fragile***.
- Able to separate the computational side of acquiring and managing context information with the compositional side of developing pervasive computing application.

Summary and Future Work

- Contribution
  - A framework which ***cultivates the self-coordination*** of pervasive computing environments.
  - A ***programming abstraction*** that eases the development of applications.
- Outstanding Issues
  - Further validation
    - ***Expressiveness*** of the YABS .
    - System wide behaviors.
  - Context information
    - Defining an ontology of context information
  - Privacy
    - Context information kept within ***local environment***.
    - Entities have ***full control*** over what information is prorogated.
    - May need to look at other methods of securing the information.
- Questions?