Ambient Talk
in multiple networks

Tomohiro Suzuki
Belgium?
10 weeks stay in Vrije Universiteit Brussels

Brussels, Belgium
Language: French, Dutch
AmbientTalk 2

- Being developed by the members in Vrije Universiteit Brussels

- Dynamic-typed interpreter Language, Implemented in Java
  
  • Google Android!

- Service discovery abstraction

```
export: (object: {
    def hello() {"Hello!"}
}) as: A;
```

```
when: A discovered: { |a|
    when: (a<->hello()) becomes: { |r|
        println(r);
    }
}
```
Session / Pi-calculus

a session can be passed to another session
Functional Programming Language
a function can be passed to another function

Session / Pi-calculus
a session can be passed to another session
AmbientTalk?
a far reference can be passed to another far reference?

Functional Programming Language
a function can be passed to another function

Session / Pi-calculus
a session can be passed to another session
AmbientTalk?

a far reference can be passed to another far reference?

It works!

As long as they are in the same network

Every AmbientTalk VM knows other VMs’ IP address and port number as VM ID.
AmbientTalk?
a far reference can be passed to another far reference?

It works!
As long as they are in the same network

Every AmbientTalk VM knows other VMs’ IP address and port number as VM ID.
AmbientTalk VM that belong to two or more networks

Every AmbientTalk VM knows other VMs’ IP address and port number as VM ID.
AmbientTalk VM that belong to two or more networks

Every AmbientTalk VM knows other VMs’ IP address and port number as VM ID.
Multi-network branch

svn/interpreter/branches/multinetwork
svn/atlib/branches/multinetwork

Tomohiro Suzuki

• Port object
• Gateway Actor
• export: as: on: function
• Virtual Port Object
• Next?
Port Objects

To control two or more network interfaces

Interactive AmbientTalk Shell, version 2.18 (development)
>def ports := networks.getAll()
>>[<native object: port(en0/192.168.2.1:offline)>,
    <native object: port(en1/134.184.43.198:offline)>]
>def p1 := ports[1]
>>(<native object: port(en0/192.168.2.1:offline)>
>pl.online()
>>nil
>pl
>>(<native object: port(en0/192.168.2.1:51296[AmbientTalk])>

Now AmbientTalk VM can belong to more than one networks
Before that...

Interactive AmbientTalk Shell, version 2.15 [build160410]
>network.online()
>>nil
>

This makes the VM to use *an* interface that has an IP address
Gateway Actor

export: (object: {
    def hello() {"Hello!"}
}) as: A;

when: A discovered: {
    when: (a<-hello()) becomes: {
        println(r);
    }
}
export: (object: {
    def hello() {"Hello!"}
}) as: A;

when: A discovered: <|a|
gatewayOf: a created: <|gw|
export: gw as: A;
}
}

when: A discovered: <|a|
when: (a<-hello()) becomes: <|r|
println(r);
}}
Gateway Actor

```javascript
export: (object: {
    def hello() {"Hello!"}
}) as: A;
```

when: A discovered: {
    ```javascript
    when: (a<-hello()) becomes: {
        println(r);
    }
    ```
}

gatewayOf: a created: {
    ```javascript
    export: gw as: A;
    ```
}

when: A discovered: {
    ```javascript
    when: (a<-hello()) becomes: {
        println(r);
    }
    ```
}
Network X

Gateway Actor

```javascript
export: (object: {
  def hello() {
    "Hello!"
  }
}) as: A;

when: A discovered: { |a|
  gatewayOf: a created: { |gw|
    export: gw as: A;
  }
}
```

Network Y

```javascript
when: A discovered: { |a|
  when: (a<-hello()) becomes: { |r|
    println(r);
  }
}
```
def gatewayOf: a created: handler {
    /* creates an actor, grabs a dummy object
       and passes it to handler */
    def gw := actor: { |a|
        def dummyObj := object: {};
        def initialize() {
            def m := reflectOnActor();
            m.becomeMirroredBy: (extend: m with: {
                /* pub      *      sub
                A  <=  B  <=  C */
                def receive(receiver, msg) {
                    if: (receiver == dummyObj) then: { // Message from C to A
                        def futureFromC := msg.future;
                        def futuredMsg := super^createMessage(msg.selector, msg.arguments, []);
                        when: (a <+ futuredMsg) becomes: { |val|
                            futureFromC<-resolveWithValue(val)@[/.at.lang.firstclassrefs.MetaMessage,
                                                        /.at.lang.futures.OneWayMessage];
                        } catch: { |exp|
                            futureFromC<-ruinWithException(exp)@[/.at.lang.firstclassrefs.MetaMessage,
                                                        /.at.lang.futures.OneWayMessage];
                        }
                    } else: { // Message from A to C, to resolve futures
                        super^receive(receiver, msg);
                    }
                }
                dummyObj;
            }
        }
        when: (gw<-initialize()) becomes: { |dummy|
            handler(dummy);
        }
    }
}

import /.at.discovery.gateway;
when: A discovered: { |a|
    gatewayOf: a created: { |obj|
        export: obj as: A;
    }
}
Gateway Actor Examples

/* A controls hop counts */
when: A discovered: {|a|
  when: (a<-leftHopCount()) becomes:{|c|
    if: (c <= 0) then: {
      gatewayOf: a created: {|gw|
        export: gw as: A;
      }
      export: service as: A;
    }
  }
...
Gateway Actor Examples

/* A controls hop counts */
when: A discovered: { |a|
  when: (a<-leftHopCount()) becomes:{ |c|
    if: (c <= 0) then: {
      gatewayOf: a created: { |gw|
        export: gw as: A;
    }
  }
  gatewayOf: a created: { |gw|
    export: gw as: A;
  }
...}

/* Only services that have a certain key */
def service := object:{
def c := 3;
def leftHopCount() { c--;
}
def service := object:
  def c := 3;
def leftHopCount() { c--;
}
..."
Gateway Actor Examples

/* A controls hop counts */
when: A discovered: { |a|
  when: (a<-leftHopCount()) becomes:{ |c|
    if: (c <= 0) then: {
      gatewayOf: a created: { |gw|
        export: gw as: A;
      }
    }
  }
  export: service as: A;
...
A Middle Node Moving Networks

c.f. TOTA (tuple spaces)
A Middle Node Moving Networks

c.f. TOTA (tuple spaces)
A Middle Node Moving Networks

A Disconnection

B gateway

C

pub

sub
c.f. TOTA (tuple spaces)
A Middle Node Moving Networks

A

Disconnection

B

gateway

msg

C

pub

sub

c.f. TOTA (tuple spaces)
A Middle Node Moving Networks

c.f. TOTA (tuple spaces)
A Middle Node Moving Networks

Disconnection

Reconnection

c.f. TOTA (tuple spaces)
export: obj as: tags on: ports

GreenWifiNet

purpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

PurpleEtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet

Purple EtherNet
export: obj as: tags on: ports

D discovers A’s service two times...

portGreen.online();
portPurple.online();
when: A discovered: {{|a|
gatewayOf: a created: {{|gw|
  export: gw as: A
}}}}
We can use the port objects to specify which network to export a service.

```javascript
export: obj as: tags on: ports

portGreen.online();
portPurple.online();
when: A discovered: {|a|
  gatewayOf: a created: {|gw|
    export: gw as: A on: [portPurple];
  }
}

This still have problem: if B' is in both networks...
```
Export: as: on: Examples

when: A discovered: \{|a|\n  def portA := networks.getByFarRef(a);
def otherPorts := networks.getAll().filter:|p| p !\= portA;
gatewayOf: a created: \{|obj|\n    export: obj as: A on: otherPorts;
  }\n}
Virtual Port

logical networks among AmbientTalk virtual machines in one computer

Interactive AmbientTalk Shell, version 2.18
>def foo := networks.createPort("foo")
>>><native object: vport(foo/127.0.0.1:offline)>
>def bar := networks.createPort("bar")
>>><native object: vport(bar/127.0.0.1:offline)>
>foo.online(); bar.online();
>>nil
>foo
>>><native object: vport(foo/127.0.0.1:51944 [AmbientTalk])>
>export: obj as: A on: [bar];

Virtual Ports are distinguished by their names
They can be used as normal network port objects
Multi-network Problem

- Multiway problem

Object identity or publication identity is required
c.f. Multiway reference (Kevin's)
Multi-network Problem

- Multiway problem

Object identity or publication identity is required

C.f. Multiway reference (Kevin's)
Demo

- [http://youtube.com/suztomo](http://youtube.com/suztomo)

Note: this is solved by Multiway Reference / Kevin Pinte!
Multiway Problem
Multiway Reference in RFID
Summary: what I did

- Modified AmbientTalk VM to have several network interfaces by port object
- implemented Gateway Actor that make possible to communicate between different networks without modifying pub / sub code
- implemented port object abstraction: virtual port object
- Introduced multiway problems in the gateways
- Applied multiway reference to the problems
From Here?

- More fine-grained implementation for port object
  - currently port object configures VM
  - it should configure actors
- Routing Protocols on Gateway Actors based on speed, traffic, robustness, etc?

Dries Harnie is trying to implement Bluetooth stack of AmbientTalk
One more thing

• AmbientTalk mode in Emacs
• Syntax highlight
• Indentation