Ambient References
Addressing Objects in Mobile Networks

Tom Van Cutsem

Programming Technology Laboratory
Vrije Universiteit Brussel
Mobile Networks

Pervasive/Ubiquitous Computing
Mobile Networks

Pervasive/Ubiquitous Computing
Mobile Networks

Pervasive/Ubiquitous Computing

cellPhone#debit(price)
vendingMachine#getProduct(id)
Mobile Networks

Pervasive/Ubiquitous Computing

Ambient References
“remote actor references”

cellPhone#debit(price)
vendingMachine#getProduct(id)
Mobile Networks

Pervasive/Ubiquitous Computing

Ambient References
“remote actor references”

Ad Hoc P2P Networks

Volatile Connections
Service description \rightarrow \textbf{?} \rightarrow \textbf{remote reference}

\begin{itemize}
  \item p2p discovery?
  \begin{itemize}
    \item concurrency control
    \item callbacks partition code
    \item managing disconnections
  \end{itemize}
\end{itemize}

\begin{verbatim}
... Discovery.search(serviceDescription,
  new DiscoveryListener() {
    void foundService(Service s) {
      // use the service
    }
    void lostService(Service s) {
      // manage disconnection
    }
  });
\end{verbatim}
Resilience

- Temporary disconnections
  - should not break a remote reference
  - should not immediately raise exceptions
- Communication should resume upon reconnection
Resilience

- Temporary disconnections
  - should not break a remote reference
  - should not immediately raise exceptions
- Communication should resume upon reconnection
Resilience

- Temporary disconnections
  - should not break a remote reference
  - should not immediately raise exceptions
- Communication should resume upon reconnection
Transitory Addressing

- Remote references: UID-based, often device-dependent
- Too inflexible: cannot rebind
Transitory Addressing

- Remote references: UID-based, often device-dependent
- Too inflexible: cannot rebind
Transitory Addressing

- Remote references: UID-based, often device-dependent
- Too inflexible: cannot rebind

//192.168.0.101/
InfoBooth@0x015a54f76

//192.168.0.102/
InfoBooth@0x75a24f7ff
Group Communication

- Abstract from multitude of devices
- Ad hoc ‘proximate’ groups

collections?

discovery manager
add/delete
iterate
client

services

concurrency control
boilerplate iteration code
Problem Statement

- Standard remote object references fail to meet these requirements
- Need for dedicated referencing abstractions for mobile networks
Computational Model

• Services are ‘public’ **actors** advertising themselves via *service types*

```plaintext
servicetype InstantMessenger;

actor {
    provide(InstantMessenger);
    method talk(text) { ... };
}
```
• Services are ‘public’ actors advertising themselves via service types

```java
servicetype InstantMessenger;

actor {
    provide(InstantMessenger);
    method talk(text) { ... };
}
```
Ambient References

- Two states: **bound** or **unbound**
- Binds to proximate matching services

```
aMessenger = ambient InstantMessenger
aMessenger#talk("Hello")
```
Ambient References

- **Two states**: *bound* or *unbound*
- Binds to proximate matching services

```
aMessenger = ambient InstantMessenger
aMessenger#talk("Hello")
```
Ambient References

- Two states: **bound** or **unbound**
- Binds to proximate matching services

```plaintext
aMessenger = ambient InstantMessenger
aMessenger#talk("Hello")
```
Ambient References

- Two states: **bound** or **unbound**
- Binds to proximate matching services

```plaintext
aMessenger = ambient InstantMessenger
aMessenger#talk("Hello")
```
Two states: bound or unbound

Binds to proximate matching services

```java
aMessenger = ambient InstantMessenger
aMessenger#talk("Hello")
```
Design Dimensions

• Design family of remote references
• each suitable for different kind of collaboration
• Three properties:
  • Scope of binding
  • Elasticity
  • Cardinality
Scope of binding
Scope of binding

```
servicetype iDrive;
```
Scope of binding

```plaintext
servicetype iSign;

servicetype iDrive;
```
Scope of binding

```
servicetype iSign;
servicetype iSell;
servicetype iDrive;
```
Scope of binding

servicetype iSell;
servicetype iSign;
servicetype iDrive;
servicetype iAm;
Scope of binding

```
servicetype iSell;

servicetype iSign;

servicetype iAm;

servicetype iDrive;

aCar = ambient iDrive;
```
Scope of binding

```
servicetype iAm;
servicetype iSign;
servicetype iDrive;
servicetype iSell;

aShop = ambient iSell;
```
Scope of binding

```
servicetype iAm;
servicetype iSign;
servicetype iSell;
servicetype iDrive;

name = ...;
forSale = ...;
discount = ...;

aShop = ambient iSell;
```
Scope of binding

```
servicetype iAm;
servicetype iSign;
servicetype iSell;
servicetype iDrive;
```

```
name = ...;
forSale = ...;
discount = ...;
```

```
aShop = ambient iSell s where
    s.forSale.includes(“gizmo”);
```
Elasticity

servicetype iSell;
servicetype iSign;
servicetype iDrive;
servicetype iAm;
Elasticity

```
servicetype iSell;
servicetype iSign;
servicetype iDrive;
```
Elasticity

customer = ambient(1 day) iAm;
Elasticity

```
servicetype iAm;

servicetype iSign;

servicetype iDrive;

servicetype iSell;

favoriteShop = ambient! iSell s
    where s.name = "...";
```
Cardinality

```
servicetype iSell;

servicetype iSign;

servicetype iAm;

servicetype iDrive;
```
Cardinality

`servicetype iSell;`

`servicetype iSign;`

`servicetype iDrive;`

`servicetype iAm;`
Cardinality

```
servicetype iSell;
servicetype iSign;
servicetype iDrive;

nearbyCars = ambient[2] iDrive;
```
Cardinality

```
servicetype iSell;
servicetype iSign;
servicetype iDrive;

nearbyCars = ambient* iDrive;
```
## Taxonomy

### Scope of binding

<table>
<thead>
<tr>
<th>Elasticity x Cardinality</th>
<th>Fragile</th>
<th>Elastic</th>
<th>Sturdy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uni</td>
<td>ambient S;</td>
<td>ambient(e) S;</td>
<td>ambient! S;</td>
</tr>
<tr>
<td>Multi</td>
<td>ambient[n] S;</td>
<td>ambient(e)[n] S;</td>
<td>ambient![n] S;</td>
</tr>
<tr>
<td>Omni</td>
<td>ambient* S;</td>
<td>ambient(e)* S;</td>
<td>ambient!* S;</td>
</tr>
</tbody>
</table>
Requirements revisited

Provisionality

Resilience

Transitory Addressing

Group Communication
Requirements revisited

Provisionality

Resilience

Transitory
Addressing

Group
Communication

aService = ambient ServiceType

Scope of binding
Requirements revisited

Provisionality

Resilience

Transitory Addressing

Group Communication

aService = ambient! ServiceType

Elasticity
Requirements revisited

Provisionality

Resilience

Transitory Addressing

Group Communication

aService = ambient(t) ServiceType

Elasticity

Scope of binding
Requirements revisited

- Provisionality
- Resilience
- Transitory Addressing
- Group Communication

\[ \text{aService} = \text{ambient}^* \text{ ServiceType} \]
• Local **proxy** for remote service
• Performs discovery on behalf of its client
Implementation

- Local **proxy** for remote service
- Performs discovery on behalf of its client
Conclusion

• Pervasive computing requires novel language abstractions!

• Ambient references: remote object references for mobile networks