Distribution and Mobility in Software Architectures

José Luiz Fiadeiro
Research Context

Software Architecture
— foundations
— design primitives
— evolution

Mathematical Framework
Category Theory

Prototype Language
CommUnity

Joint work with
— Antónia Lopes (Univ. Lisbon, Portugal)
— Michel Wermelinger (UNL, Portugal)
— Luís Andrade
Software Architecture

Structural Models

Components

Connectors
CommUnity: Designs

CommUnity designs are defined in terms of a **signature**

— **channels** (input, output, private) and **action names** (shared, private)

and **actions**

**Example.** A design that models a producer

```
design producer is
out x:nat
prv rd: bool
do prod:[rd [] x::nat||rd:=false]
[] send:[rd [] rd:=true]
```
CommUnity: Interaction and Composition

design cable is
in 1:nat
do ac:true]\ skip

x][i][x
send][ac] rec

design producer is
out x:nat
prv rd: bool
do prod:[rd [] x:[]nat|rd:=false]
 send:[[]rd [] rd:=true]

x][i][x
send][ac] rec

design consumer is
in x:nat
prv y: nat, rdc: bool
do rec:[rdc [] y:=x || rdc:=false]
[] use:[[]rdc [] rdc:=true]
Motivation

Abstracted
— the way the components are distributed over a network
— the way the connections among the nodes of the network are realized

Assumed
— computations are performed in fixed hosts
— hosts are linked by static connections
— all forms of coordination are considered to be realizable
Motivation

Abstracted
— the way the components are distributed over a network
— the way the connections among the nodes of the network are realized

Assumed
— computations are performed in fixed hosts
— hosts are linked by static connections
— all forms of coordination are considered to be realizable

Mobile computing in ad-hoc networks
new forms of dist.systems
Distributon and Mobility in SA

Architectures for Mobility
FET/IST project, Global Computing Initiative
LMU, DIPISA, DSIUF, IEI-CNR, ATX, FFCUL, IinfUW, ULEICES

Goal
to develop na architectural approach in which mobility aspects can be modelled explicitly as part of the application domain and mapped into the distribution and communication topology made available at physical levels
Motivation

Architecture-based approaches

Computation

Coordination
CommUnity designs are defined in terms of a **signature**

- **channels** (input, output, private) and **action names** (shared, private)

- **location variables** (input, output)

**Position where the value is available**

\[ x@l \]

\[ g@\{l_1,l_2,...\} \]

**Position where code is executed**

\[ g@l \]

\[ D(g) \rightarrow G(g) \rightarrow R(g) \]

**enabling condition**

**write frame**

**state changes**
CommUnity with Distribution: **Designs**

**Example.** A design that models a producer — a centralised system that does not control its own movement

```plaintext
design producer is
inloc lp
out x@lp:nat
prv rd@lp: bool
do prod@lp: [rd [] x:[]nat||rd:=false]
[] send@lp: [rd [] rd:=true]
```
Another example. A design that models a distributed system that partially controls its own movement

design prod-cons is
inloc lp
outloc lc
out x@lp:nat
prv y@lc:nat, rdp@lp,rdc@lc: bool
do prod@lp:[rup @lc: [rup y:=x|rdp:=false]]
trans@lp:[rup @lc: [rup y:=x|rdp:=false]]
move@lc:[rup @lc: [rup y:=nattoloc(y)|rdc:=true]]
CommUnity with Distribution:

**Externalisation of Superposed Behaviour**

- prod-cons
- producer $\rightarrow$ sync $\rightarrow$ consumer
  - externalisation of communication
  - externalisation of distribution and mobility

- producer $\rightarrow$ sync $\rightarrow$ consumer$'$ $\rightarrow$ mov
CommUnity with Distribution:

Externalisation of Distribution

design cable is
inloc 1
in i:nat
do  ac:true[] skip

design consumer' is
inloc lr
in x
out y@lr:nat
do rec@lr:[true[] y:=x]

mov

design mov is
outloc 1
in y@1:nat,
prv rd@1: bool
do  action@1:[rd[] rd:=false]
[] move@1:[[]rd@1:=nattoloc(y)
||rd:=true]
Conclusions

- Architectural techniques based on the separation of computation and coordination can be applied to mobile systems;
- Distribution and Mobility patterns can be explicitly represented within architectural models through distribution connectors;
- Distribution Connectors can also be used for modelling new forms of coordination that have emerged in mobile computing
  - coordination styles that are location-dependent Transient Interaction
  - coordination styles that involve the management of the locations of the coordinated parties Remote Evaluation
Future Work

- Abstract a **General Mathematical Characterisation** of the properties underlying Distributed CommUnity, in order to make these ideas available to other formalisms and development platforms.

- Extend with the new distribution dimension the **coordination technologies** developed by ATX for managing system evolution.