Engineering Self-organising Coordination
A Computational Field Approach

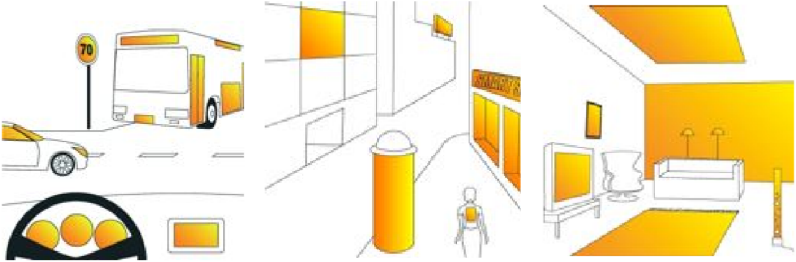
Mirko Viroli

Dipartimento di Informatica: Scienza e Ingegneria (DISI)
Alma Mater Studiorum—Università di Bologna, Italy
mirko.viroli@unibo.it

COORDINATION 2013, New Directions Session, Firenze, 3/6/2013

Outline

Tomorrow living and computational environments



(Pict. by A. Ferscha, JKU)

Impact on the computational model

Towards new levels of complexity on:

- Scale: devices are very dense (forming a continuum)
- Situatedness: data/events are tightly linked with space
- Distribution: most situations and events are distributed
- Opennes: highly unpredictable working conditions
- Mobility: continuous relocation, new deployments in the short-term

Requirements for coordination models

- Adaptation: services should face unexpected distributed contingencies
- Emergence: no full design-time awareness of system behaviour

⇒ Self-organisation: creation of "function" with no global supervision

... what we call in one word: Self-organising Coordination

Impact on the computational model

Towards new levels of complexity on:

- Scale: devices are very dense (forming a continuum)
- Situatedness: data/events are tightly linked with space
- Distribution: most situations and events are distributed
- Opennes: highly unpredictable working conditions
- Mobility: continuous relocation, new deployments in the short-term

Requirements for coordination models

- Adaptation: services should face unexpected distributed contingencies
- Emergence: no full design-time awareness of system behaviour

⇒ Self-organisation: creation of "function" with no global supervision

.. what we call in one word: Self-organising Coordination

Impact on the computational model

Towards new levels of complexity on:

- Scale: devices are very dense (forming a continuum)
- Situatedness: data/events are tightly linked with space
- Distribution: most situations and events are distributed
- Opennes: highly unpredictable working conditions
- Mobility: continuous relocation, new deployments in the short-term

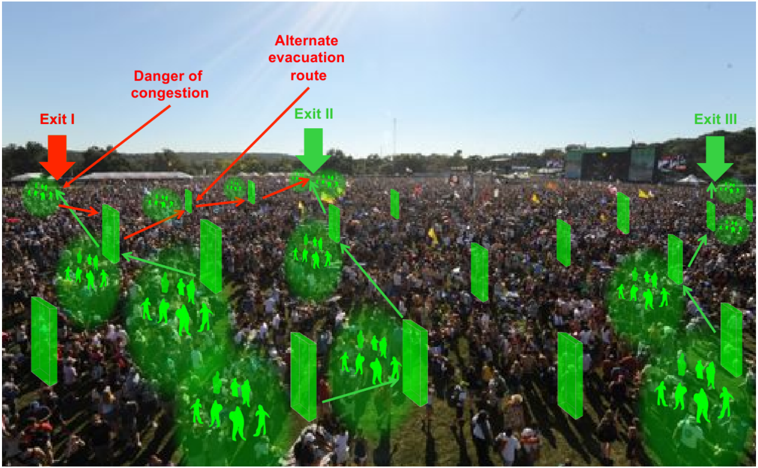
Requirements for coordination models

- Adaptation: services should face unexpected distributed contingencies
- Emergence: no full design-time awareness of system behaviour

$\Rightarrow$ Self-organisation: creation of "function" with no global supervision

.. what we call in one word: Self-organising Coordination

Crowd Engineering – www.sapere-project.eu



(Pict. by A. Ferscha, JKU)

An increasingly interesting business for coordination

Advanced coordination mechanisms

- Coordinating sensing/acting in complex situations
- Opportunistic exploitation of contextual information
- Dealing with user (implicit) preferences, intentions, social network
- Anticipation of future contingencies
- Spontaneous aggregation/composition of data/functions

Some application scenarios

- Pervasive computing (crowd steering, coordinated visualisation)
- Sensor networks (aggregating/diffusing data, complex recognition)
- Swarm and modular robotics (cooperative team work)

An increasingly interesting business for coordination

Advanced coordination mechanisms

- Coordinating sensing/acting in complex situations
- Opportunistic exploitation of contextual information
- Dealing with user (implicit) preferences, intentions, social network
- Anticipation of future contingencies
- Spontaneous aggregation/composition of data/functions

Some application scenarios

- Pervasive computing (crowd steering, coordinated visualisation)
- Sensor networks (aggregating/diffusing data, complex recognition)
- Swarm and modular robotics (cooperative team work)

Outline

1. Emerging Pervasive Computing Scenarios and Coordination
2. Self-organising Coordination by Computational Fields
3. Some recent result
4. Conclusions

Self-organisation meeting space-based coordination

Going to the aggregate: from tuples to fields of tuples

- Tuples are pointwise abstractions, reifying single events/data
- Shall move to "distributed groups" of tuples as first-class abstraction
  - Best known as computational fields, or co-fields
  - Created by nondeterministic aggregation/diffusion processes
  - Acting as "global self-adapting and self-organising" structures

Previous works and attempts

- TOTA middleware for co-fields [Mamei and Zambonelli, 2009]
- Biochemical TSs, $\sigma\tau$-Linda [Viroli et al., 2012, Viroli and Casadei, 2009, Viroli et al., 2011]
- SAPERE framework [Zambonelli et al., 2011]

⇒ still lacking a solid stack of models, langs, patterns, platforms, apps

Self-organisation meeting space-based coordination

Going to the aggregate: from tuples to fields of tuples

- Tuples are pointwise abstractions, reifying single events/data
- Shall move to "distributed groups" of tuples as first-class abstraction
  - Best known as computational fields, or co-fields
  - Created by nondeterministic aggregation/diffusion processes
  - Acting as "global self-adapting and self-organising" structures

Previous works and attempts

- TOTA middleware for co-fields [Mamei and Zambonelli, 2009]
- Biochemical TSs, $\sigma\tau$-Linda
  [Viroli et al., 2012, Viroli and Casadei, 2009, Viroli et al., 2011]
- SAPERE framework [Zambonelli et al., 2011]

$\Rightarrow$ still lacking a solid stack of models, langs, patterns, platforms, apps

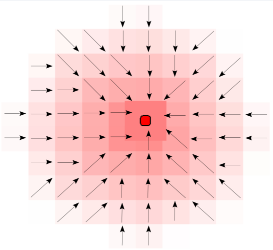
The gradient case

The gradient field as a key building block

Spreading: $x \Rightarrow x \,|\, (x+1)^{\rightsquigarrow}$ *Aggregation:* $x \,|\, y \Rightarrow min(x, y)$

- In each node minimal distance to the source (where $x = 0$) stabilises
- Usages: advertising, routing, relocation, distributed aggregation

Gradient shape (discrete, flat continuous, structured continuous)



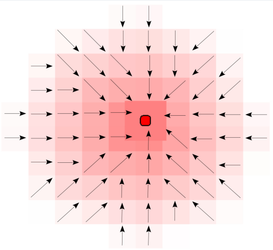
The gradient case

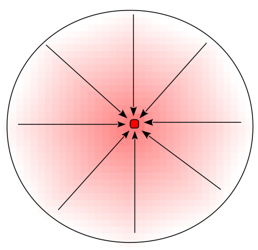
The gradient field as a key building block

Spreading: $x \Rightarrow x \mid (x+1)^{\rightsquigarrow}$ *Aggregation:* $x \mid y \Rightarrow min(x,y)$

- In each node minimal distance to the source (where $x = 0$) stabilises
- Usages: advertising, routing, relocation, distributed aggregation

Gradient shape (discrete, flat continuous, structured continuous)





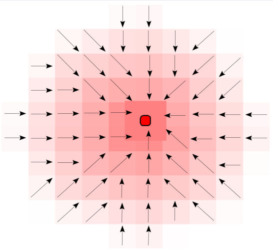
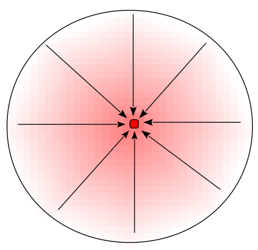
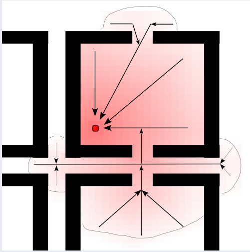
The gradient case

The gradient field as a key building block

Spreading: $x \Rightarrow x \,|\, (x+1)^{\rightsquigarrow}$ *Aggregation:* $x \,|\, y \Rightarrow min(x,y)$

- In each node minimal distance to the source (where $x = 0$) stabilises
- Usages: advertising, routing, relocation, distributed aggregation

Gradient shape (discrete, flat continuous, structured continuous)

Layers in engineering computational fields

Low level – single-device capabilities: aggregation, diffusion, decay,..

- Theory: component/interaction models
- Practice: supporting platforms

Mid level – constructs to express/compose fields

- Theory: formal languages and properties
  (self-stabilisation, expressiveness, convergence with increasing density)
- Practice: coordination languages, libraries of patterns

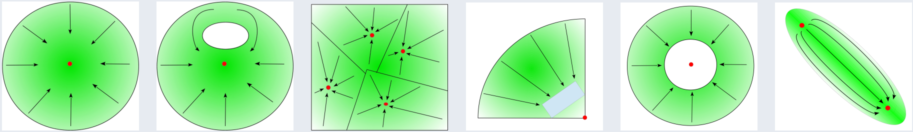
Top level – perception/actuation over fields: chemotaxis, morphogenesis

- Theory: usage patterns, comprehensive languages, agent models
- Practice: information routing, crowd steering, distributed recognition

Outline

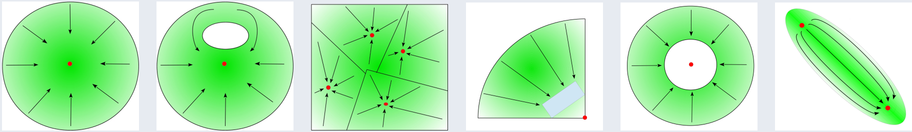
Early results (1/3): Towards a catalogue of patterns

A set of patterns [Montagna et al., 2013]

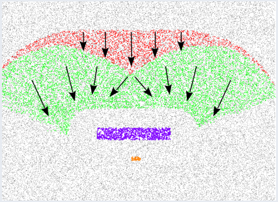


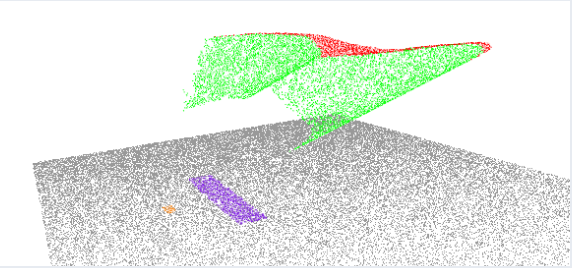
Early results (1/3): Towards a catalogue of patterns

A set of patterns [Montagna et al., 2013]



Anticipative gradient [Montagna et al., 2012]





Early results (2/3): Models/languages to express fields

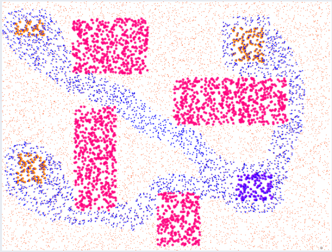
Several proposals in the past

- See a survey in [Beal et al., 2013] – few of them in coordination
- Trying to balance expressiveness and tractability [Viroli, 2013]

A recent core calculus: "[{..} ..]"

```
def If(FIELD,WHERE) is
     WHERE ? FIELD : #undef
def Gossip(VAL) is
     [{VAL}]
def Grad(SRC,OBS) is
     [If({SRC}+1,not(OBS))]
def Dist(SRC,DST,OBS) is
     Gossip(If(Grad(SRC,OBS),DST))
def Chan(S,D,O,W) is
     Grad(S,O) + Grad(D,O) + W < Dist(S,D,O)
```

..and the resulting "channel"



Early results (3/3): Confluence and Self-Stabilisation

The problem of formal properties

- Predict/control the final outcome of a field computation
- Find sufficient conditions for certain key properties – self-stabilisation

Generating confluent fields [Viroli, 2013]

Considering the fields expressed in the language above

- Theorem: they all are confluent and self-stabilising

⇒ Sufficient conditions

1. There is a noetherian order relation over data
2. The functions connecting "[]" and "{}" are monotonic and progressive
3. No recursive calls

Early results (3/3): Confluence and Self-Stabilisation

The problem of formal properties

- Predict/control the final outcome of a field computation
- Find sufficient conditions for certain key properties – self-stabilisation

Generating confluent fields [Viroli, 2013]

Considering the fields expressed in the language above

- Theorem: they all are confluent and self-stabilising

⇒ Sufficient conditions

1. There is a noetherian order relation over data
2. The functions connecting "[]" and "{}" are monotonic and progressive
3. No recursive calls

Outline

Conclusions

The direction of "self-organising coordination"

- Self-org and coordination will inevitably and increasingly combine
- Engineering self-organisation is challenging..
- ..coordination people can contribute a lot
- Coordination fields as a candidate first-class abstraction

Perspectives

- Self-org coordination can target many objectives of Horizon 2020
- It can be a good ingredient of proposals under different umbrellas
  - ⇒ adaptation, autonomic computing, collective systems, future ICT, SmartCities, Internet of things, Socio-technical systems

Conclusions

The direction of "self-organising coordination"

- Self-org and coordination will inevitably and increasingly combine
- Engineering self-organisation is challenging..
- ..coordination people can contribute a lot
- Coordination fields as a candidate first-class abstraction

Perspectives

- Self-org coordination can target many objectives of Horizon 2020
- It can be a good ingredient of proposals under different umbrellas
  - ⇒ adaptation, autonomic computing, collective systems, future ICT, SmartCities, Internet of things, Socio-technical systems

Acknowledgements

The SAPERE FP7 Project

Many ideas presented here have been bootstrapped in the context of the EU-FP7-FET Proactive project (Franzo Zambonelli as leader)

SAPERE

Self-Aware Pervasive Service Ecosystems, under contract no. 256873.

CINA Italian PRIN Project

In the next 3 years CINA Project in Italy will provide a container for nurturing some tasks of this research thread (Rocco de Nicola as leader)

CINA

Compositionality, Interaction, Negotiation, Autonomicity for the future ICT society

References I

Beal, J., Dulman, S., Usbeck, K., Viroli, M., and Correll, N. (2013).
Organizing the aggregate: Languages for spatial computing.
In Mernik, M., editor, *Formal and Practical Aspects of Domain-Specific Languages: Recent Developments*, chapter 16, pages 436–501. IGI Global.
A longer version available at: `http://arxiv.org/abs/1202.5509`.

Mamei, M. and Zambonelli, F. (2009).
Programming pervasive and mobile computing applications: The tota approach.
ACM Transactions on Software Engineering and Methodologies, 18(4).

Montagna, S., Pianini, D., and Viroli, M. (2012).
Gradient-based self-organisation patterns of anticipative adaptation.
In *Proceedings of 6th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2012)*, pages 169–174. IEEE.

References II

Montagna, S., Viroli, M., Fernandez-Marquez, J. L., Di Marzo Serugendo, G., and Zambonelli, F. (2013).
Injecting self-organisation into pervasive service ecosystems.
Mobile Networks and Applications, 18(3):398–412.

Viroli, M. (2013).
Engineering confluent computational fields: from functions to rewrite rules.
In *Spatial Computing Workshop (SCW 2013)*, AAMAS 2013, Saint Paul, Minnesota, USA.

Viroli, M. and Casadei, M. (2009).
Biochemical tuple spaces for self-organising coordination.
In *Coordination Languages and Models*, volume 5521 of *LNCS*, pages 143–162. Springer.

Viroli, M., Casadei, M., Montagna, S., and Zambonelli, F. (2011).
Spatial coordination of pervasive services through chemical-inspired tuple spaces.
ACM Transactions on Autonomous and Adaptive Systems, 6(2).

References III

Viroli, M., Pianini, D., and Beal, J. (2012).
Linda in space-time: an adaptive coordination model for mobile ad-hoc environments.
In *Coordination Languages and Models*, volume 7274 of *LNCS*. Springer-Verlag.

Zambonelli, F., Castelli, G., Ferrari, L., Mamei, M., Rosi, A., Serugendo, G. D. M., Risoldi, M., Tchao, A.-E., Dobson, S., Stevenson, G., Ye, J., Nardini, E., Omicini, A., Montagna, S., Viroli, M., Ferscha, A., Maschek, S., and Wally, B. (2011).
Self-aware pervasive service ecosystems.
Procedia CS, 7.