Engineering Adaptive Service Ecosystems

Mirko Viroli
mirko.viroli@unibo.it

ALMA MATER STUDIORUM—Università di Bologna, Cesena

Academic Year 2012/2013

"Self-Aware Pervasive Service Ecosystems"

SAPERE European Project

- European call for funding projects: FP7-ICT-2009.8.5
- Call name: Self-awareness in Autonomic Systems
- Period: 1/10/2010 - 30/9/2013
- Consortium: (i) Università di Modena e Reggio Emilia, Italy
  (ii) Università di Bologna, Italy
  (iii) University of Geneve, Switzerland
  (iv) University of St Andrews, Dublin, UK
  (v) Johannes Kepler Universitaet Linz, Austria

Impact on the Cesena site

- A "fulcrum" of challenging EU-wide research activities,
- An opportunity for students (projects, thesis, post-laurea)

Outline/Goals of this seminar

- Deepen the "Future Pervasive Computing" scenario
- Describe a bit of the SAPERE Project
- Present main research ideas/results/challenges
- Explain a bit of self-organisation via computational fields
- Propose collaborations

Outline

Pervasive Computing

A definition

Pervasive computing is a model of computation in which information processing has been thoroughly integrated into everyday objects and activities, namely, into the environment in which humans live. In the course of ordinary activities, someone "using" pervasive computing engages some computational devices, and may not necessarily even be aware of it.

Synonims, possibly with slightly different acceptations

- Ubiquitous computing
- Internet of Things (Focus on the Web)
- Ambient Intelligence (Focus on "house environments")
- Smart-cities (Focus on "city-level environments")

Challenges of Future (and emerging) Pervasive Computing

1a — Increasing introduction of pervasive devices

(Pict. by A. Ferscha, JKU)

Challenges of Future (and emerging) Pervasive Computing

(Pict. by A. Ferscha, JKU)

Challenges of Future (and emerging) Pervasive Computing

(Pict. by A. Ferscha, JKU)

Challenges of Future (and emerging) Pervasive Computing

(Pict. by A. Ferscha, JKU)

Challenges of Future (and emerging) Pervasive Computing

1 — Increasing introduction of pervasive devices

- The world will be more and more populated by computing devices
  - tablets, smart phones, tags, cameras, displays, sensors, actuators, watches, glasses
- Note that the number of such devices follows an exponential law
  - Fact: since 2009 we have more connected devices than humans
  - Fact: today, less than 1% of devices are connected
  - Fact: the number of devices and connected devices grows exponentially
  - Conjecture: ..they will keep growing exponentially
- The network will become a true virtual counterpart of the world
  - concerning events, sociality, business, logistics, physical structure
  - it will be increasingly dense up to be understood as a continuum
  - with obvious ethic and social consequences, which I won't deepen here..

Challenges of Future (and emerging) Pervasive Computing

2 — Production/consumption of large masses of data and services

- Devices will produce large masses of data
- Users will themselves increasingly inject their data in the system
  - Facebook, Youtube (Dropbox)
- We will expect that any environment properly and promtply reacts to the situation
  - to our profile, preferences, requirements, social obligations
- Not just data, we are already facing universes of applications
  - Android Market may become a standard deployment tool

Challenges of Future (and emerging) Pervasive Computing

3 — Opennes, self-*, contex-awareness will be mandatory

- Opennes: we won't know which services, data, users, devices will be available soon, the infrastructure should work independently of this
- Self-* features should naturally emerge
  - Self-adaptiveness: tuning behaviour to ongoing changes
  - Self-organisation: finding a spatio/temporal global structures
  - Self-optimisation: be able to garbage unneeded services/data
  - Self-awareness: intrinsic identification of situations
- Context-awareness: data and services will be relative to the position/location in which they reside
  - centralisation of data and software might be abandoned, because of privacy and performance
  - (or, will clouds take over?)

The challenge of Pervasive Computing

Oppurtunities for industry

- We are far from supporting what we have in mind
- Only specific solutions to specific problems so far (even by academia)
- Opportunistic interaction of devices very hard currently

A lot of work is going on

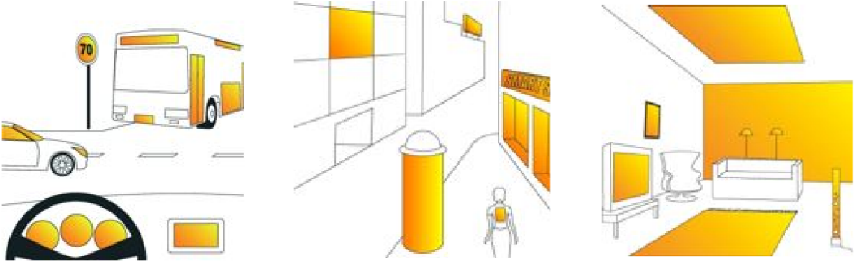
- Research contexts: SOA, P2P, Grid, Cloud, Self-org, Coord
- Argue there are *two next big things* in Information Technology
  - Pervasive Computing
  - Bio-ICT convergence (e.g. nature-inspired computing)

Outline

The scenario of display ecosystems

(Pict. by A. Ferscha, JKU)

Displays are (and will increasingly be) pervasive



(Pict. by A. Ferscha, JKU)

Some interesting services and features

- Displays show information based on majority of people around
- Alerts are shown as a given person passes nearby
- Content redirected from smartphone to display
- Adjacent displays show a common, bigger content
- Displays coordinate to avoid irritating users
- Displays coordinate to provide visualisation streams
- Public displays sharedly used by friends to interact
- Using eye-glasses for immersed interaction

User-Display Coordination

(Pict. by A. Ferscha, JKU)

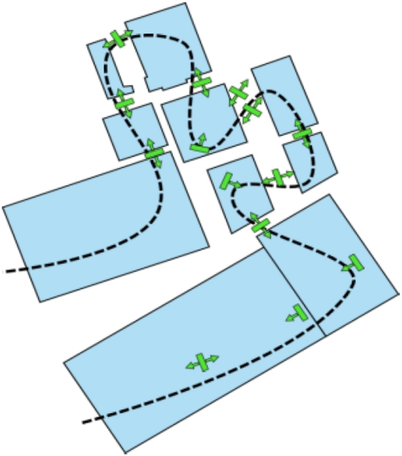
Enhancing displays

(Pict. by A. Ferscha, JKU)

Displays Coordination

(Pict. by A. Ferscha, JKU)

Content Coordination

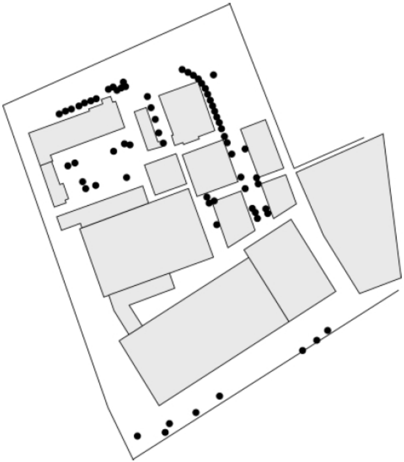


(Pict. by A. Ferscha, JKU)

Single-user steering

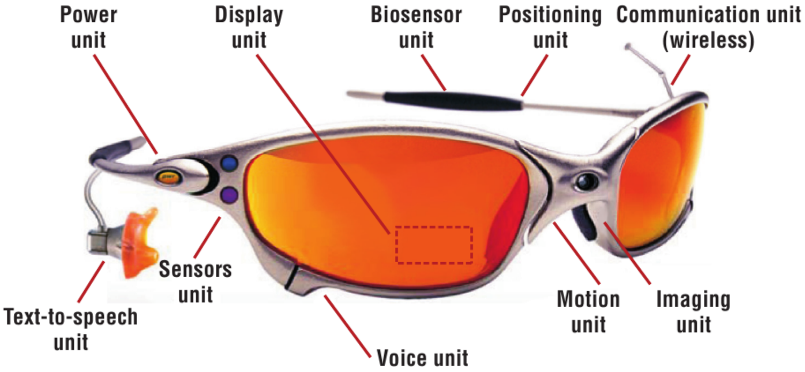
(Pict. by A. Ferscha, JKU)

Crowd steering in complex environments



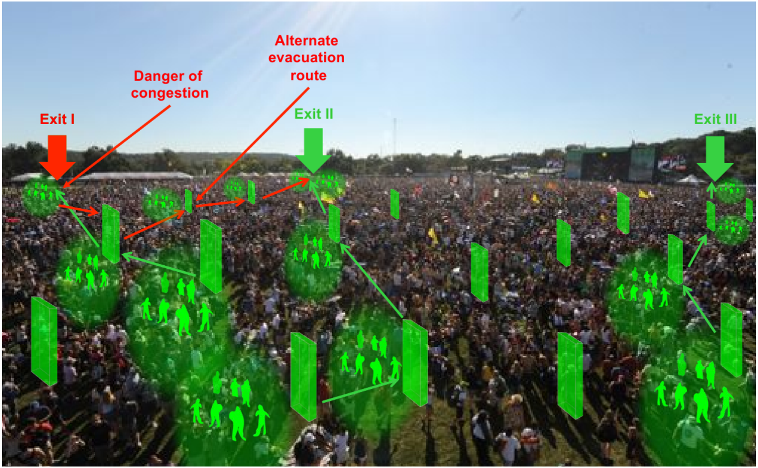
(Pict. by A. Ferscha, JKU)

New displays



(Pict. by A. Ferscha, JKU)

The goal of crowd engineering



(Pict. by A. Ferscha, JKU)

Crowd-based application scenario

Other cases

- Single User Steering via Public Displays
- Crowd Balancing
- Evacuation
- Adaptive Advertisement
- Collective Guidance

Not just displays

Why this emphasis on display-based applications?

Displays are paradigmatic!

- Displays are the quintessential embedded system
- A display as a node with enhanced input/output abilities
- Smartphones, tablets, smart watches, *are* smart displays
- Displays are useful, interesting commercially, and unexploited

⇒ Displays as the key of pervasive computing

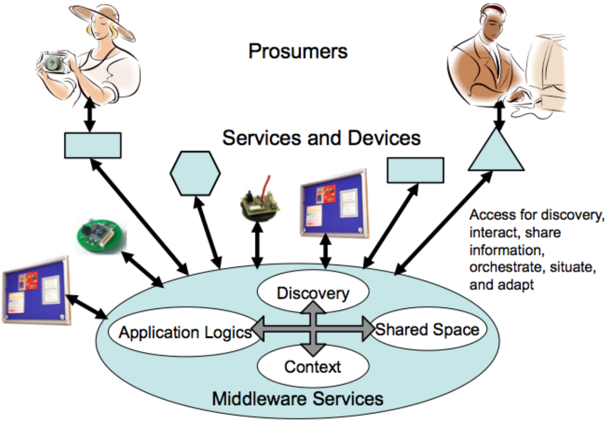
Other applications of the proposed techniques

- Sensor networks (gathering environmental info)
- Mobile robotics (drones, collective robotics)

Abstract of SAPERE

- Development of a theoretical and practical framework for deployment and execution of self-aware and adaptive services;
- applications to future and emerging pervasive network scenarios;
- getting inspiration from natural ecosystems,
- modelling and deploying services as autonomous individuals in an ecosystem of other services, data sources, and pervasive devices,
- enforcing self-awareness and autonomic behaviours as inherent properties of the ecosystem.

A standard, centralised SOA solution – cloud oriented



A standard, centralised SOA solution

A centralised solution

One service (typically in the cloud) for:

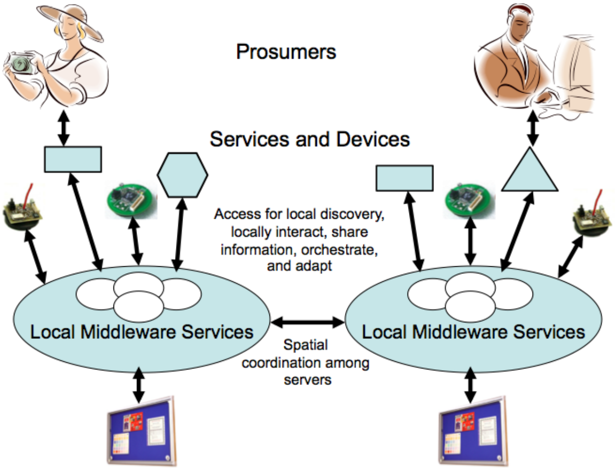
- Discovery: what components are available in the system?
- Context: where are components? (behaviour specialisation)
- Orchestration: coordinating components
- Data storage: depositing/retrieving data
- Adaptation: reacting to contingencies

All components interact through such middleware services

Does it scale well?

- Can displays promptly react?
- Can we store streams of sensor data in the cloud?
- What about privacy?

De-centralising the SOA solution

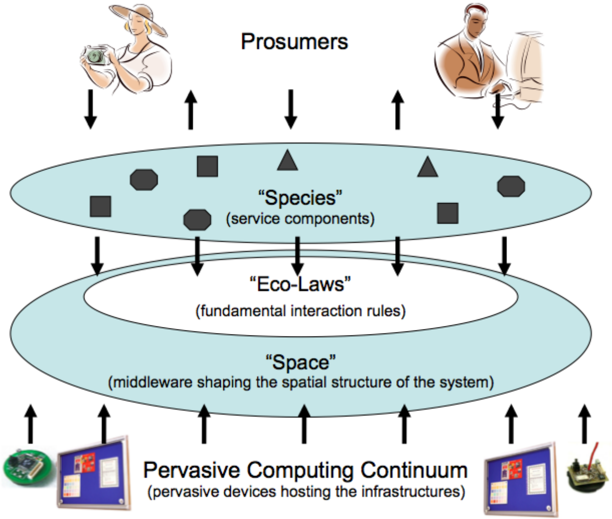


De-centralising the SOA solution

Duplicating middleware services in each location

- Better deals with situated components
- Simplifies contextualisation, discovery, and orchestration
- The role of shared spaces becomes more important
- Adaptation is still complex and crucial
- Might go to the cloud for a subset of functions

Eco-inspired SOA solution



Eco-inspired SOA solution

Fully decentralised middleware services

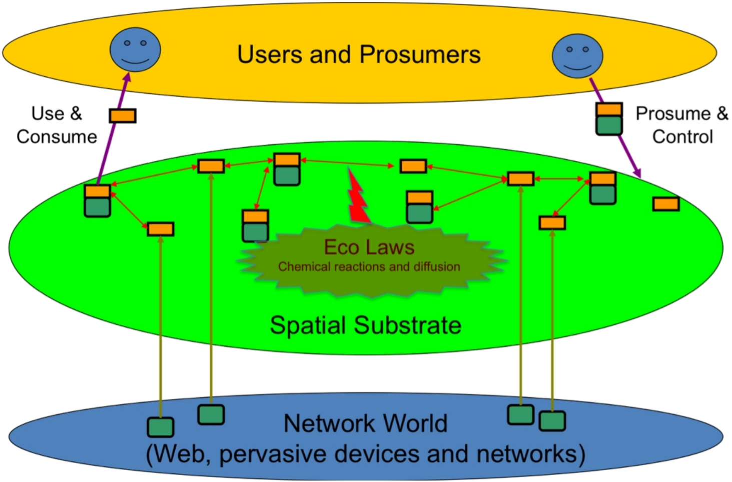
- Locations become very small and form a huge dynamic set
- Contextualisation, discovery, and orchestration almost vanish
- Middleware service just as a single space distributed on all nodes
- In overall we have a network of spaces with service "tags"
- Adaptation is achieved by simple rules combining tags

Drawing a bridge with natural ecosystems

We have a set of spatially situated entities interacting according to well-defined set of natural laws enforced by the spatial environment in which they situate, and adaptively self-organizing their interaction dynamics according to their shape and structure[a]

[a] Franco Zambonelli and Mirko Viroli. "A survey on nature-inspired metaphors for pervasive service ecosystems". In: *International Journal of Pervasive Computing and Communications* 7.3 (2011), pages 186–204. ISSN: 1742-7371.

Abstract Architecture



Pervasive Ecosystems[1]

Example Patterns

Data/devices/services are added by injecting their tags, then:

- they connect to other tags, supporting agent-to-agent interaction
- they could diminish until their population extinguishes
- they compete with other "species", and may survive
- they compose with patches injected to improve them
- they diffuse around
- they move where the context is favourable
- they aggregate with other copies, forming an overlay

[1] Franco Zambonelli et al. "Self-aware Pervasive Service Ecosystems". In: *Procedia CS* 7 (2011), pages 197–199.

Eco-laws and Live Semantic Annotations

Live Semantic Annotations (LSA)

- A unified description for devices, data, services
- Is about interface, status, and behaviour of a component
- It provides semantic information, and it is dynamic

Eco-Laws

- They resemble chemical reactions
- They take some reagent LSA, and provide some product LSA
- They can diffuse an LSA in the neighborhood
- They can aggregate LSAs like in chemical bonding
- They form a small & fixed set of natural eco-laws

Outline

What is SAPERE

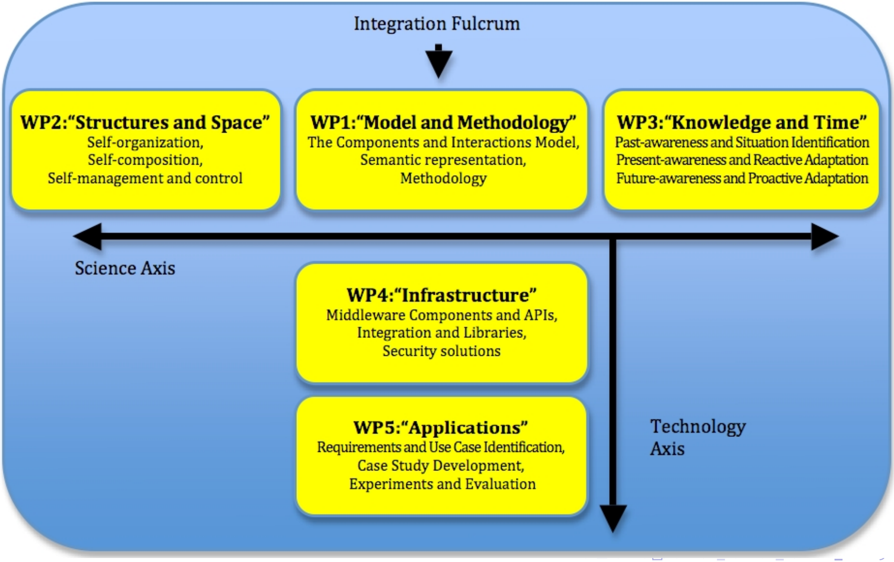
A European Project

- European call for funding projects: FP7-ICT-2009.8.5
- Call name: Self-awareness in Autonomic Systems
- Call type: Future and Emerging Technologies (FET)
- Funding Scheme: STREP (Specific Targeted Research Project)
- Period: 1/10/2010 - 30/9/2013
- Overall research grant: $\approx$ 2.5MEuro
- Consortium
  - Università di Modena e Reggio Emilia – Franco Zambonelli
  - Università di Bologna – Mirko Viroli
  - University of Geneve – Giovanna di Marzo
  - University of St Andrews, Dublin – Simon Dobson
  - Johannes Kepler Universitaet Linz – Alois Ferscha

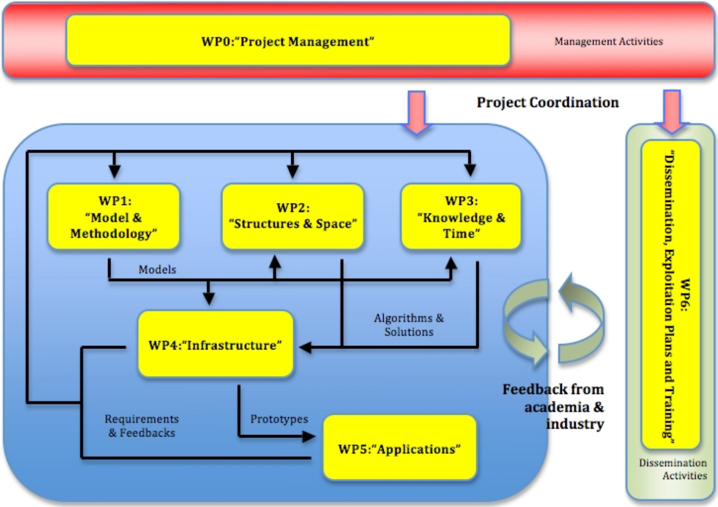
Elements of a Project Document (in general)

- Abstract
- Objectives
- Novelty and Contribution w.r.t. existing works
- Workplan (division in tasks, timing, efforts)
- Technical descriptions
- Deliverables
- Measure success indicators
- Dissemination activities
- Financial aspects

Strategy


Integration Fulcrum
WP2:"Structures and Space"
Self-organization,
Self-composition,
Self-management and control
WP1:"Model and Methodology"
The Components and Interactions Model,
Semantic representation,
Methodology
WP3:"Knowledge and Time"
Past-awareness and Situation Identification
Present-awareness and Reactive Adaptation
Future-awareness and Proactive Adaptation
Science Axis
WP4:"Infrastructure"
Middleware Components and APIs,
Integration and Libraries,
Security solutions
WP5:"Applications"
Requirements and Use Case Identification,
Case Study Development,
Experiments and Evaluation
Technology
Axis

Activities


WP0:"Project Management"
Management Activities
Project Coordination
WP1: "Model & Methodology"
WP2: "Structures & Space"
WP3: "Knowledge & Time"
Models
Algorithms & Solutions
WP4:"Infrastructure"
Requirements & Feedbacks
Prototypes
WP5:"Applications"
Feedback from academia & industry
WP6: "Dissemination, Exploitation Plans and Training"
Dissemination Activities

WP1: Model & Methodology – Leader UNIBO

Task T1.1 – The Components and Interaction Models

- Abstract (representation independent) model of eco-laws
- Services structure and interactions
- Studying analysis tools for behaviour verification

Task T1.2 – Semantic representation

- Shape of LSAs
- Studying analysis tools for logic reasoning

Task T1.3 – Methodology

- Finding a SE methodology
- Conceiving tools for development/analysis

WP1: Deliverables

D1.1 – Early Operational Model (M12) – Editor: UNIBO

- First version of abstract computational model
- First version of live semantic annotation framework
- Early demonstration of the operational model at work

D1.2 – Complete Operational Model (M20) – Editor: UNIBO

- ...

D1.3 – Final Operational and Semantic Model (M24) – Editor: UNIBO

- ...

D1.4 – Early Report on Methodology (M24) – Editor: STA

- ...

D1.5 – Methodology and Analysis Suite (M32) – Editor: UNIBO

- Final methodological guidelines
- Description of associated analysis tools

D1.6 – Final Report on Methodology and Suite (M36) – Editor: UNIBO

- Complete and refined engineering methodology documentation
- Assessment of methodology and of associated tools.

Efforts

Partic. no.	Partic. short name	WP0	WP1	WP2	WP3	WP4	WP5	WP6	Total person months
1	UNIMORE	**15**	12	0	22	**34**	16	2	101
2	BIRKBECK	1	15	**36**	0	21	0	1	74
3	STA	1	13	8	**36**	0	15	1	74
4	UNIBO	1	**36**	20	0	20	0	4	81
5	JKU	1	0	16	16	0	**40**	2	75
Total		19	76	80	74	75	71	10	405

UNIBO details

- Leader of WP1: “Model and Methodology”
- Leader of WP6: “Dissemination, Exploitation, ..”
- Involved in WP2: “Structures & Space”
- Involved in WP4: “Infrastructure”

The global life-cycle of a project

Before

- Preparation of preliminary results
- Creation of a consortium
- Writing the proposal
- Continuing do so until getting a project accepted

Meanwhile

- Achieve (at least some) of the envisioned results
- Fight a bit with the partners
- Strive to create reasonable deliverables

After

- Keep creating those results
- Bootstrapping a new project

Outline

Key Features – and corresponding architectural elements

Toleration of diversity: Live Semantic Annotations (LSA)

- Each individual (agent) is represented by one (or more) LSAs
- They uniformly reflect/affect the agent inner state

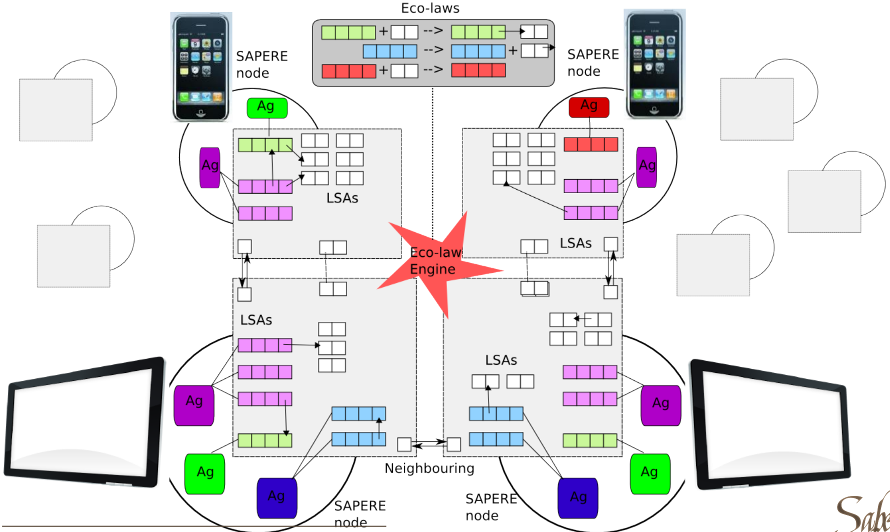
Situatedness: Spaces and Bonding

- LSAs are spread in LSA-spaces, each hosted by a device
- LSAs can connect to other via bonds, allowing mutual observation

Adaptivity: Eco-laws

- They create/change/delete/move LSAs, promoting agent interaction
- They can enact self-organisation, i.e., structures robust to changes

Operational Architecture[2]



[2] Mirko Viroli, Danilo Pianini, Sara Montagna, and Graeme Stevenson. "Pervasive Ecosystems: a Coordination Model based on Semantic Chemistry". In: *27th Annual ACM Symposium on Applied Computing (SAC 2012)*. (SAC). Edited by Sascha Ossowski, Paola Lecca, Chih-Cheng Hung, and Jiman Hong. Riva del Garda, TN, Italy: ACM, 2012. ISBN:

A natural metaphor

We chose chemistry as a reference metaphor

- LSAs as chemicals, eco-laws as reactions
- LSA-spaces as "ambients" supporting diffusion
- All tightly bridged with Semantic Web Technologies (RDF/SPARQL)

Main ingredients

Situation	Agents are situated in SAPERE nodes
Action	Agents manifest through local LSAs
Context	LSAs can bond to others around
Observation	An agent observes the world via bonds from his LSAs
Reaction	Eco-laws manipulate LSAs semantically
Diffusion	Eco-laws can relocate LSAs

LSAs (serialised as RDF[3])

Structure of LSAs

- A pair of a unique LSA-id, and a semantic description (SD)
- SD as a set of multi-valued (ontology-based) properties, as in RDF
- Overall: an RDF-like graph (Resource Description Framework)
    - RDF as a set of subject, predicate, object triples
    - Strings or URIs (Uniform Resource Identifiers)

Example: The LSA of a crowd-sensor in a museum scenario

```
lsa:crowdsensorlsa1123
    eco:#time "2011-05-30T11:00:00";
    eco:#loc sid:node34164@room132;
    eco:type museum:crowdsensor museum:contextlsa;
    museum:sensingtime "2011-05-30T11:31:24";
    museum:crowdlevel "0.93";
```

[3] *RDF Primer*. http://www.w3.org/TR/rdf-primer/. 2011.

Eco-laws

Basic facts

- Chemical rules over LSA patterns: `P+...+P --r--> Q+...+Q`
- Patterns mean pre-/post-conditions, applied at rate *r*
- Patterns are template of LSAs
  - Constrained variables written `?V(filter)`
  - Can check values for presence (keyword "+", assumed by default), absence ("-"), unique existence ("=")

Triggering a display ?DIS visualisation of ?ADV because of ?USR

```
?DIS eco:type museum:display; museum:status ="ready";  +
?ADV eco:type museum:ad; museum:content ?C;            +
?USR eco:type museum:usr; museum:profile ?P(?P matches ?C);
--r-->
?DIS museum:status ="showing"; museum:service ?ADV;
+ ?ADV + ?USR
```

Eco-laws to SPARQL[4]

Triggering a display ?DIS visualisation of ?ADV because of ?USR

```
?DIS eco:type museum:display; museum:status ="ready";  +
?ADV eco:type museum:ad; museum:content ?C;            +
?USR eco:type museum:usr; museum:profile ?P(?P matches ?C);
--r-->
?DIS museum:status ="showing"; museum:service ?ADV;
+ ?ADV + ?USR
```

Equivalent SPARQL(/SPARUL) formulation

```
SELECT DISTINCT * WHERE{
  ?DIS eco:type museum:display .
  ?DIS museum:status "ready" .  FILTER NOT EXISTS { ?DIS museum:status ?o . FILTER (?o!= "ready") }
  ?ADV eco:type museum:ad .
  ?ADV museum:content ?C .
  ?USR eco:type museum:user .
  ?USR museum:profile ?P .      FILTER(?P rdf:type ?C) .
}
REMOVE DATA {!DIS museum:status ?o}
INSERT DATA {!DIS museum:status "showing"}
INSERT DATA {!DIS museum:service !ASV"}
```

[4] *ARQ - A SPARQL Processor for Jena.* http://jena.sourceforge.net/ARQ/. 2011.

Motivations for using RDF/SPARQL

For the users

- RDF is a standard and open language for describing resources
- RDF can be relative to widely-used ontologies
- Ontologies allow one to check LSAs for correctness

For implementers

- Parsers for RDF/SPARQL already exist
- SPARQL-enabled engines exist[a], usable for scheduling/firing laws
- RDF/Ontologies intrinsically support forms of semantic matching

[a] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. "Pellet: A practical OWL-DL reasoner". In: *Web Semant.* 5 (2 2007), pages 51–53. ISSN: 1570-8268.

Achieving context-awareness and self-organisation in SAPERE

The two key ingredients

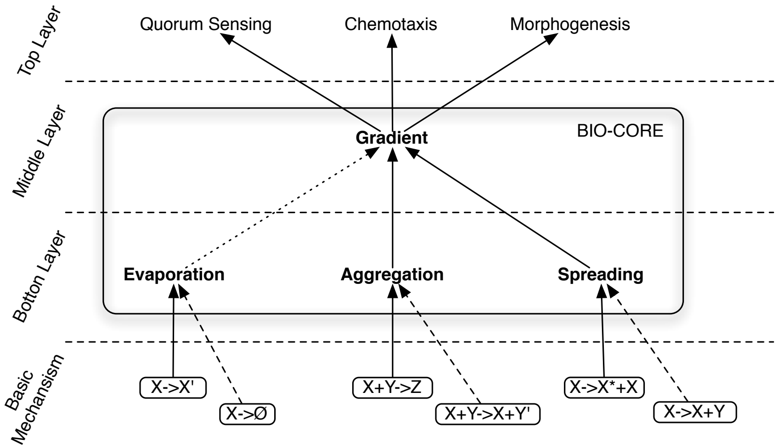
1. Bonding/matching for local sensing/acting/interaction
2. Aggregation/spreading/decaying for (global) self-organisation

A computational field-based approach

- Agents advertise services or requests by creating computational fields, i.e., distributed and self-stabilising structures of LSAs
- Remote agents bond with field LSAs and observe source information

⇒ All this supported by eco-laws or by library agents available in each node

Self-organisation Patterns[5]



[5] Jose Luis Fernandez-Marquez, Giovanna Di Marzo Serugendo, Sara Montagna, Mirko Viroli, and Josep Lluis Arcos. "Self-Organising Design Patterns". In: *Natural Computing* (2012). Edited by G. Rozenberg and H.P. Spaink. To appear. ISSN: 1567-7818.

Aggregation, Decay, and Spreading eco-laws

Triggered by agents simply by injecting LSAs with specific properties

Aggregration: out of two (/many) compatible LSAs, one is retained

LSA triggering properties: *aggregate* (ODI function) and *type* (target property)

Decay: make an LSA vanish

LSA triggering properties: *decay* (numerical deadline value)

Spreading: sends a copy of the LSA to all neighbours (or one)

LSA triggering properties: *diffusion_op* (propagation kind), and optional *destination*, *type*, *previous*.

Computational fields

A way to engineer self-organising applications in different scenarios

- Sensor networks (aggregating/diffusing data)[a]
- Swarm and modular robotics (cooperative team work)[b]
- Pervasive computing (crowd steering, coordinated visualisation)[c]

---

[a] Jacob Beal and Jonathan Bachrach. "Infrastructure for Engineered Emergence on Sensor/Actuator Networks". In: *IEEE Intelligent Systems* 21.2 (2006), pages 10–19. ISSN: 1541-1672.

[b] Jonathan Bachrach, Jacob Beal, and James McLurkin. "Composable continuous-space programs for robotic swarms". In: *Neural Computing and Applications* 19.6 (2010).

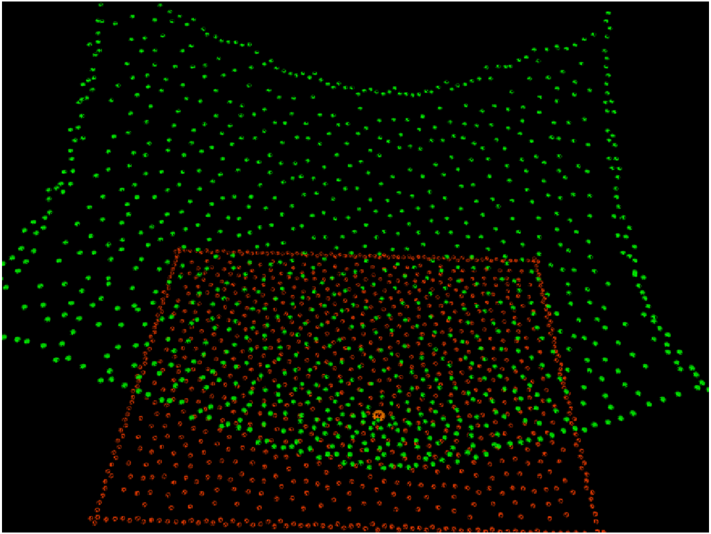
[c] Mirko Viroli, Matteo Casadei, Sara Montagna, and Franco Zambonelli. "Spatial Coordination of Pervasive Services through Chemical-inspired Tuple Spaces". In: *ACM Transactions on Autonomous and Adaptive Systems* 6.2 (2011), 14:1 –14:24.

Definition

- A function ϕ mapping nodes of a (dense) network to structured values
- Defines a spatio-temporal structure situated in the (physical) world
- Paradigmatic case: a *gradient* (i.e., `distance-to` field)

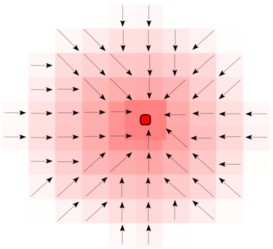
$$x \Rightarrow (x+1)^{\rightsquigarrow} \qquad x+y \Rightarrow min(x,y) \qquad x \Rightarrow \epsilon$$

A 3D rendering of a uniform gradient structure



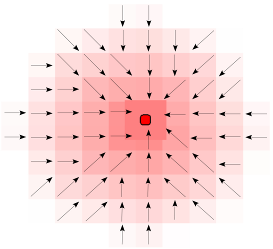
Towards a library of gradient-based computational fields

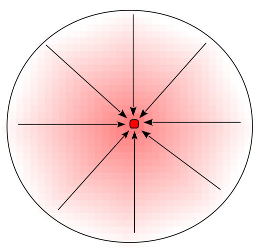
Gradient (discrete, flat continuous, structured continuous)



Towards a library of gradient-based computational fields

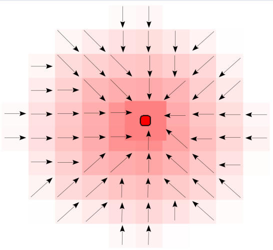
Gradient (discrete, flat continuous, structured continuous)

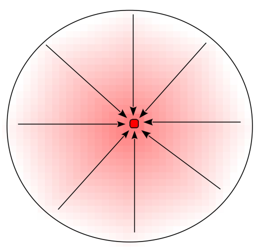


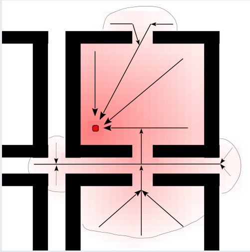


Towards a library of gradient-based computational fields

Gradient (discrete, flat continuous, structured continuous)

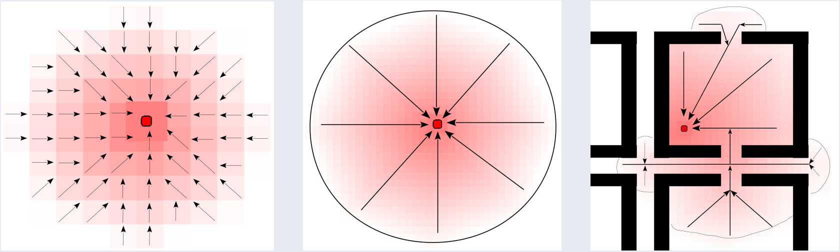




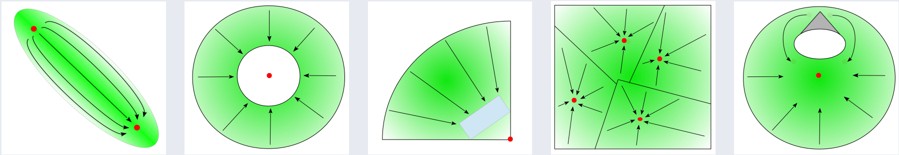


Towards a library of gradient-based computational fields

Gradient (discrete, flat continuous, structured continuous)



Other structures (channel, shrinking crown, sector, partition, shadow, . . .)

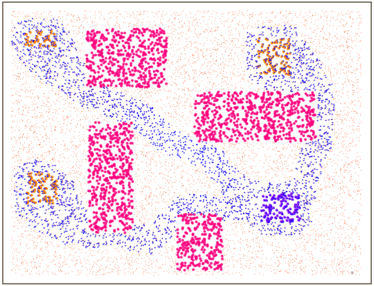


Example 1: Creating optimal (multi-)paths

Idea and implementation in SAPERE

Enable devices used to steering from a source to a target

- Gradients are spread from source S and destination D
- D also spreads the perceived distance from S
- The 3 gradients are used in each device to decide what is enabled

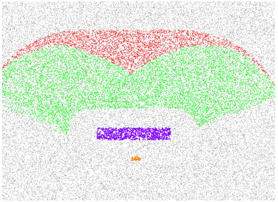


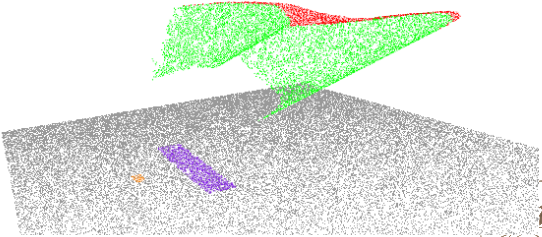
Example 2: Anticipative gradient [SASO 2012]

Idea and implementation in SAPERE

Anticipating the effect of obstacles known to appear in the future

- Gradients are spread from target T and future obstacle O (in $[T_1, T_2]$)
- Changes occur where "*sector*(T,O)" and "*crown*(O, T_1, T_2)" overlap
- There, either circumvent O (green) or proceed and wait (red)





Related Works

Programmable Tuple Spaces

TuCSoN[a], MARS

⇒ We aim at more carefully balancing expressiveness and tractability

[a] Andrea Omicini and Franco Zambonelli. "Coordination for Internet Application Development". In: *Autonomous Agents and Multi-Agent Systems* 2.3 (Sept. 1999), pages 251–269.

Self-organisation in Tuple Spaces

TOTA[a], SwarmLinda

⇒ We achieve self-organisation by laws of reaction/diffusion

[a] Marco Mamei and Franco Zambonelli. "Programming pervasive and mobile computing applications: The TOTA approach". In: *ACM Trans. Softw. Eng. Methodol.* 18.4 (2009), pages 1–56.

Spatial Computing

The Proto Language, `http://proto.bbn.com/`

Outline

SAPERE and students

Opportunities

- Small Projects (for this exam, for LMC)
- Master Thesis
- Master Thesis to a project partner
- Post-degree trip to a partner
- Post-degree collaboration

Master thesis so far

Completed

- Tosi 12/2010 – Architecture of LSA-space
- Virruso 12/2010 – Eco-laws and fuzzy matching
- Santarelli 03/2011 – LSA-space with eco-laws
- Pianini 03/2011 – Simulator for self-organisation
- Desanti 10/2011 – Complete LSA-space Alpha
- Pronti 10/2011 – Complete Simulator Alpha
- Morgagni 10/2011 – Android Integration
- Cioffi 07/2012 – Self-organisation patterns
- Contessi 07/2012 – Semantic LSA-space
- Polverelli 03/2013 – Gradient-based fields

New opportunities

Macro-areas

- Developing applications
    - Kinect integration, smartphones detection
- Middleware implementation
    - there will be need of a new lightweight and modular middleware
- Designing new self-organisation patterns
    - advanced steering strategies (composition, anticipation)
- Developing Alchemist simulator
    - Go to next seminar