A Coordination Approach to Spatially-Situated Pervasive Service Ecosystems

Mirko Viroli\textsuperscript{1} Elena Nardini\textsuperscript{1} Gabriella Castelli\textsuperscript{2} Marco Mamei\textsuperscript{2} Franco Zambonelli\textsuperscript{2}

\textsuperscript{1}Alma Mater Studiorum – Università di Bologna 
\{mirko.viroli, elena.nardini\}@unibo.it
\textsuperscript{2}Università di Modena e Reggio Emilia 
\{gabriella.castelli, marco.mamei, franco.zambonelli\}@unimore.it

WOA 2011
Rende, July 5, 2011
1 Referring Context and Aim

2 A Natural-inspired Approach

3 An Use Case

4 Conclusion and Future Work
Increasing Introduction of Pervasive Devices

- Increasing introduction of pervasive devices (e.g. PS, PDAs, smart phones, tags, displays, etc.)
- Prosumption of large masses of data and services (e.g. from users and devices)
- Software will grow increasingly and be an “eternal beta”
- Opennes, self-* and context-awareness will be mandatory
Recent proposals for pervasive computing scenarios try to account for issues related to:

- **Spatiality** [Mamei and Zambonelli, 2009, Murphy et al., 2006]
- **Spontaneous and opportunistic** [Autili et al., 2009, Fok et al., 2009]
- **Self-adaptation and self-management** [Roy et al., 2008]

In most of cases, the proposals face these issues via one-of solutions to specific problems in specific areas

→ **Lack of generality and comprehensiveness**

→ **Idea:** Re-thinking current service architectures and coordination approaches by taking inspiration from natural systems where spatial concepts and self-* features are inherently there because of the basic “rules of the game”
The Logic Architecture

**General features**

- A spatial substrate mapped above the actual pervasive network infrastructure composed by:
  - A dynamic set of computational spaces spread in the pervasive computing environment
  - Components (devices, users and software services) taking part in the ecology
- Each component has associated a semantic representation called Live Semantic Annotation (LSA)
- LSA-spaces store LSAs
- Eco-laws drive the dynamics of the ecosystem – i.e., coordinate individual activities – as sort of chemical reactions among LSAs
Four levels
Component/Agent Model

Agents and their LSAs

- An agent (implicitly/explicitly) injects one (or more) LSAs
- It does so in the local LSA-space
- Quitting/moving to be handled on top by some service
- Can read/write owned LSAs, can read also LSAs it is bonded to

Interaction primitives

- inject: takes prop descriptions, yields LSA-id
- observe: takes LSA-id and prop names, yields values (also by notif.)
- update: takes LSA-id and some new prop descriptions
- remove: takes LSA-id, drops whole LSA
A Natural-inspired Approach

Live Semantic Annotations (LSAs)

General features

- Semantic annotations with the same expressiveness of standard frameworks like RDF
- A unique LSA identifier $i$
- A list of $[p_i \rightarrow v_i]$ associations, where $p_i$ is the name of a property and $v_i$ is the associated value
- Constraints to the type of values associated to a property, or cardinality of a property, can be specified and enforced via OWL-like ontologies
- Some values/properties starting with symbol # are related to a special management by the infrastructure (e.g. #location is associated to the identifier of the SAPERE node it is stored in)
A Natural-inspired Approach

An Abstract Eco-law Framework I

General Features of an Eco-law

- Eco-laws drive and rule the activities of the ecosystem: concept-based (semantic and goal-oriented) networking, composition and coordination of data and services
- An eco-law is a chemical-resembling reaction transforming a reagent set $R$ into $R'$ by speed $r$: $R \rightarrow_r R'$
  - a reagent set is a sum of LSA “patterns” $P$
  - $R$ is selected based on LSAs structure and mutual matching
  - the choice of $R$ has a “rate” $r$ (useful for ranking)
Eco-law Patterns

- Can be void \((0 \rightarrow_R R)\)
- Can specify:
  - the identifier of a LSA
  - that a property either holds a term \(t\) or should be added/removed with a value \(v\)
- A term \(t\) can be:
  - a ground value \(v\)
  - a variable \(\{x\}\)
  - an annotated variable \(\{x : x f t\}\), that should match \(t\) by the function \(f\)
    - \(f\) is application dependent
    - \(f\) is a binary fuzzy predicate, yielding a real number in between 0 (no match) and 1 (full match)
Eco-law Effects

Possible effects of an eco-law:

- change values of properties in some LSAs of $R$
- removes some LSA of $R$
- creates new LSAs
- links some LSAs of $R$
- fires some LSAs of $R$ into neighbouring spaces
**The Adaptive Displays Use Case**

### Scenario

- **Public Area** (a mall, an airport) where some people wander around with their portable device (e.g. smartphone, PDA) that keep their preferences.
- Devices embed a number of sensors that may be used for inferring what the user is doing (e.g. the microphone can be used for inferring if the user is talking with someone).
- A number of public displays are deployed to show different kinds of visualisation services to users, e.g. news, advertisement and directions.
- Public displays acquire data about users standing in front of them and provide the best possible content to users.
- SAPERE nodes host LSA spaces.
- When a user is in front of a display there is an LSA-space holding: the public display’s LSA, the user’s LSA and the LSA visualisation services that the display can show.
- Eco-laws fill the context of each public display choosing the most proper visualisation service as a summary of the current audience.
The Adaptive Displays Use Case II
An Use Case

Display Context-aware Visualisation I

```json
{d}:{type=display, contextualizing=true} + {u}:{type=user}
-->[CTX-USR-IN]
{d}:{context+=u} + {u}

{d}:{type=display, contextualizing=true, screenprops={p}} +
{s}:{type=service, screenprops={p':p' matches p}}
-->[CTX-SER]
{d}:{context+=s} + {s}

{d}:{type=display, status=ready, showService={s}, showUser=+{U}} +
{s}:{content={c}} +
{t}:{type=#time, value={tt}}
-->[ACT]
{d}:{channel=#sink, status=showing, showTime={tt}, showContent={c}} +
{s}:{channel#+source} + {t} +
{l}:{type=log, time={tt}, service={s}, display={d}, showUser=+{U}}
```
Note that

- If a user or a visualisation service would become unavailable, the infrastructure automatically garbage collects their LSA and eventually drop any reference to them from other LSAs.

- The reaction manager into the LSA-space never schedules eco-laws whose neat effect is void.

- The display’s agent can decide whether contextualisation is always enabled or not, by updating LSA property `contextualising`.
Relying on a Recommendation

\[
\begin{align*}
\{d\}:&\{\text{type=display, status=ask}\} + \\
\{r\}:&\{\text{type=recommender, question=\{q':q' matches service-request\}}\} \\
\quad \longrightarrow &\{\text{ASK}\} \\
\{d\}:&\{\text{status=ask}\} + \{r\}:\{\text{pending+={d}}\} \\
\{d\}:&\{\text{type=display, status=ask}\} + \{r\}:\{\text{type=recommendation, device={d}, answerService={s}, answerUser=\star\{U\}, answerState={ss}}\} \\
\quad \longrightarrow &\{\text{REP}\} \\
\{d\}:&\{\text{status=ready, showService={s}, showState={ss}, showUser=\star\{U\}}\}
\end{align*}
\]
To support a non-local mutual awareness of displays, we rely on the concept of *computational field* [Mamei and Zambonelli, 2009] and in particular on *distributed gradient* structures of LSAs [Viroli et al., 2011]

As soon as any agent is willing to advertise its existence into a region of the network, it should inject an LSA of type *field*, with the mandatory properties:

- *source*, the agent’s LSA identifier
- *distance*, initially set to 0
- *range*, set to the maximum distance at which the agent wants to be perceived

To address the multiplicity of field LSAs produced, specific eco-laws retain LSAs diffuse along paths of shortest distance and the youngest LSAs

Eco-law reifying the diffusion mechanism needs an expressive *rate*, that is sufficiently high to guarantee prompt propagation of changes, but also sufficiently low to not bloat the network with LSA diffusion
An Use Case

Viroli et al. (UniBo-UniMoRe)
Steering the Crowd

{d}: [type=display, diffuse=true, range={r}, showService={s},
     showContent={c}, #location={l}, showTime={tt}]
--->[PUMP]
{d} +
{f}: [type=field, from={l}, source={d}, subject=display, content={c},
     range={r}, distance=0, time={tt}]

{u}: [type=user, profile={p}] +
{d}: [type=display, status=showing, context has {u}, showUser has-not {u}]
{f}: [type=field, source={d'}, subject=display, content={c: c matches p},
     from={l}] +
{n}: [type=neighbour, where={l}, direction={dd}]
--->[STEER]
{u}: [suggestion={d'}, direction={dd}] + {d} + {f} + {n}
Coordinated Visualisation

{d}:{type=display, status=showing, showUser has {u}, showState={ss}, range={r}} +
{u}:{type=user, connected=false} +
{t}:{type=#time, value={tt}}
-->[QUIT]
{d}:{showUser == {u}} + {u} + {t} +
{f}:{type=field, source={u}, subject=quitting, service={s}, range={r}, distance=0, time={tt}, showState = {ss}}

{d}:{type=display, contextualizing=true} +
{f}:{type=field}
-->[CTX-FIELD]
{d}:{context+={f}} + {f}
Conclusion and Future Work

We outlined a new coordination architecture focussing on the chemical-resembling eco-laws framework for pervasive systems.

Future work aims at:

- exploiting standard technologies like RDF for the incarnation of LSAs and eco-laws
- exploiting OWL ontologies for promoting the conception of an application-dependent set of eco-laws
- adopting more advanced spatial mechanisms into eco-laws, to support higher-level self-organising structures of LSAs
This work has been supported by the EU-FP7-FET Proactive project SAPERE – Self-aware Pervasive Service Ecosystems – under contract no. 256874.

References II


References III
