Coordinating Spatially-Situated Pervasive Service Ecosystems

Mirko Viroli, Elena Nardini ALMA MATER STUDIORUM – Università di Bologna Cesena, Italy {mirko.viroli, elena.nardini}@unibo.it Gabriella Castelli, Marco Mamei, Franco Zambonelli Università di Modena e Reggio Emilia Reggio Emilia, Italy {gabriella.castelli, marco.mamei, franco.zambonelli}@unimore.it

Abstract—Technology evolution is providing new pervasive service scenarios characterised by a huge number of distributed and dynamic devices. Accordingly, a new generation of services and infrastructures are emerging which support situatedness, adaptivity and diversity. In this paper we model the overall world of services, data and devices, as a distributed computational ecosystem. Each entity will be modelled as an autonomous, spatially-situated individual of the ecosystem, whose existence is reified by an LSA (Live Semantic Annotation). Ecosystem's behaviour is controlled by coordination rules called *eco-laws*, which are sort of chemical-like reactions evolving the population of LSAs. We describe an architecture supporting this vision, a formal model of eco-laws, and finally show their usefulness in a scenario of crowd steering through pervasive displays.

I. INTRODUCTION

The increasing evolution and spread of pervasive computing technologies is defining the basis for the emergence of a dense and global decentralised infrastructure for the creation of general-purpose pervasive services. These include traditional classes of services enriched with the capability of dynamically and autonomously adapting to the context in which they are exploited (a public display showing advertisements based on the preferences of the majority of people around), as well as innovative services for better perceiving/interacting with the physical world (a PDA hosting a real-time map showing where are my friends around and what are they doing). Such a scenario will feature a number of diverse sensing devices, personal and public displays, personal mobile devices, and humans, all of which to be dynamically engaged in very dynamic and flexible coordinated activities. In particular, the specific coordination requirements for such novel pervasive application scenarios include: (i) the capability to naturally match the spatial nature of the environment and of the services within, which involves managing coordinating activities between components that are physically co-located or close to each other; (ii) the capability to inherently facilitate spontaneous interactions among components, without requiring an a priori knowledge of each other, and making the resulting patterns of interactions self-adaptive and self-managing; (iii) the capability to flexibly tolerate evolutions of structure and diversity over time, which is necessary to account for a large (and evolving) number of very diverse components interacting with each other without being forced to face significant re-engineering to incorporate innovations and changes. To meet these above requirements, one should no longer conceive services and their coordinated activities as in ad hoc solutions or on the basis of standard serviceoriented architectures [8]. There, services are simply functional entities coordinated according to mostly static (hardly self-adaptable) patterns.

Recent proposals in the area of coordination models and middleware for pervasive computing scenarios try to account for issues related to spatiality [12], [13], spontaneous and opportunistic coordination [1], [7], self-adaptation and selfmanagement [19]. But, in most of the cases, the proposals face these issues via one-of solutions to specific problems in specific areas, and lacking generality and comprehensiveness. Tackling the problem in a more radical way, we argue that a promising direction is that of re-thinking current service architectures and coordination approaches by taking inspiration from natural systems, where spatial concepts and features of self-adaptation, self-management, and longlasting evolvability are inherently there because of the basic "rules of the game".

Nature-inspired solutions have already been extensively exploited in the area of distributed computing for the implementation of specific middleware solutions or of specific distributed services [11], and similarly, natural and ecological metaphors have been adopted to characterise the complexity of modern ICT and service systems [20]. Here we pursue a line of research that intends to go further than adopting natural system as an inspiration for specific solutions as a generic metaphor: we aim at exploiting a natural metaphor to shape a reference architecture around which to conceive, model, and develop a fully-fledged coordination framework for pervasive service systems.

Although natural systems can be categorised in different classes (e.g., physical [12], chemical [22], biological [2], or social [10]), they all account for a spatial environmental substrate, and autonomous individuals (i.e., agents) of different kinds getting in touch, interacting, competing, and combining with each other – in one word, coordinating – in respect of some basic "laws of nature". Accordingly, we claim that a truly self-adaptive shared pervasive substrate will have to be conceived as the space in which bringing to

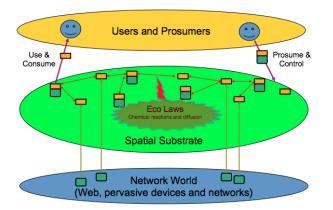


Figure 1. The SAPERE logic architecture

life an ecosystem of service agents coordinated by some basic laws (which we call "eco-laws") of the ecosystem that will provide self-organisation, self-management, and evolvability. Along these lines, the key contributions of this article are as follows:

- we frame the key concepts of the nature-inspired architecture at the basis of our coordination approach and of the SAPERE project ("Self-aware Pervasive Service Ecosystems", *www.sapere-project.eu*), in which our research activities situate (Section 2);
- we detail the model of "eco-laws" that underlies the overall dynamics of the ecosystem, and that act as spatial coordination laws over a shared distributed space (resembling a tuple space scenario), promoting spontaneous and opportunistic interactions among the components of the ecosystem (Section 3);
- we present case studies related to coordinated contextawareness and visualisation in an environment pervaded by interactive displays, which are of particular relevance for crowd steering applications (Section 4).

Related works in coordination models and languages (Section 5) and a discussion on final remarks and future works (Section 6) conclude the paper.

II. THE SAPERE COORDINATION ARCHITECTURE

The coordination approach of the proposed SAPERE model takes its primary inspiration from natural ecosystems, adopting a chemical inspired mechanism to rule the spatially-coordinated activities of the components (devices, services, and humans) and, thus, the overall system dynamics [16]. From the architectural viewpoint, we consider structuring a pervasive service environment (see Figure 1) as a non-layered *spatial substrate*, mapped above the actual pervasive network infrastructure, which is composed by a possibly very dense set of hardware devices.

The spatial substrate embeds the basic laws of nature (or *eco-laws*) that rule the activities of the system. The *components* (green blocks), living in the envisioned substrate,

will have an associated semantic representation called *Live Semantic Annotations* (LSAs, yellow blocks). To account for high dynamics of the scenario and for its need of continuous holistic adaptation, we will consider such annotations as living, active entities, capable of reflecting the current situation and context of the component they describe. They will act as actual observable interfaces of resources (devices, users, software services), as well as the basis for enforcing semantic and self-aware forms of dynamic interactions (both for service aggregation/composition and for data/knowledge management).

The eco-laws drive the dynamics of the ecosystem. We envision them to define the basic policies to rule sorts of virtual chemical reactions among the LSAs of the various individuals. In particular, the idea is to enforce dynamic concept-based (i.e., semantic and goal-oriented) networking, composition, and coordination of data and services. We will consider data and services (as represented by their associated LSAs) as sorts of chemical reagents in an ecology in which interactions and composition occur via chemical-like reactions, i.e., semantic pattern-matching between LSAs. Such reactions can contribute to establish virtual chemical bonds between entities (e.g., relating similar services with each other to produce a distributed service, or mining related data items) as well as to produce new components (e.g. a composite service coordinating the execution of atomic service components or a high-level knowledge concept derived from the aggregation of raw data items).

Coordination and adaptivity in the SAPERE framework are not bound inside the capability of individual components, but rather emerge in the overall dynamics of the ecosystem. In particular, they are ensured by the fact that any change in the system (as well as any change in its components, as reflected by dynamic changes in its LSA) will reflect in the firing of some eco-law possibly leading to the establishment of new bonds and/or in the breaking of some existing bonds. That is, the SAPERE architecture promotes adaptivity and coordination not by creating self-awareness at the level of components, but rather promoting a sort of "systemic selfawareness".

Such way of enforcing adaptation can also tolerate longterm system evolution. In fact, even if we do not assume (and do not deal with) the capability of individual components to evolve, it is the injection of new updated components in the system – automatically incorporated in the ecosystem coordination dynamics – that provide for a sort of seamless evolution. That is, as in natural selection, it is species of components that evolve, not individuals: more suitable component classes replace out-of-date ones in the ecosystem dynamics, without requiring any specific adjustment to the infrastructure or to the other components.

From an implementation-oriented viewpoint, the SAPERE framework will be realised as a lightweight and minimal middleware that will reify LSAs in the form of tuples, to

be dynamically stored and updated in a system of spatiallysituated tuple spaces spread over the devices of the network. Eco-laws will take place in the network nodes in order to promote seamless adaptivity and coordination. This architecture will also boost the participation of users that access the ecology in a decentralised way to use and consume data and services, and they can also act as prosumers by injecting new data or service components (possibly also for the sake of controlling the ecology behaviour).

III. THE ECO-LAWS FRAMEWORK

To define in a more precise way ecosystems behaviour, and shade light to the SAPERE coordination model, in this section we introduce a core calculus of eco-laws, capturing the key aspects and mechanisms of the proposed framework, which include: (*i*) syntactic structure of LSAs and eco-laws, (*ii*) application of eco-laws to LSAs (and related fuzzy matching mechanisms), (*iii*) bond establishment, and (*iv*) diffusion mechanisms to achieve spatial-aware behaviour. Although not properly a process algebra, we adopt a related formalisation approach for our calculus, as e.g. developed in [21], [4], [14]—see Section 5 for a comparison.

Syntax We model the state of an ecosystem as a networked set of LSA-spaces, each carrying a set of LSAs; each LSA is a tuple of property/value associations, and an eco-law is as a reaction (in a multiset rewrite style) over patterns of LSAs residing in the same LSA-space—with the possibility of moving LSAs to a neighbouring space, resembling so-called "firing tuples" introduced in [21].

Let meta-variable σ range over LSA-space identifiers, d over primitive data values (literals, and numerals r including ∞ – used for an arbitrary large number), x over logic variables, p over property names, type over LSA tags, f over semantic match functions, and λ over LSA unique identifiers. The (static and runtime) syntax of the model is expressed by the following grammar:

ι	::=	$\lambda \mid \lambda^{\leadsto}$	id of local/remote LSA
A	::=	$\iota: type\langle p_1 = v_1, \ldots, p_n = v_n \rangle$	LSA
v	::=	$\lambda \mid d \mid v_1 \cdot \ldots \cdot v_n$	value
S	::=	$0 \mid A \mid (S \mid S)$	LSA-space
L	::=	$R\mapsto_r R'$	eco-law
R	::=	$0 \mid P \mid R + R$	reagent set
P	::=	$0 \mid \iota \mid type \mid \langle p \ op \ t \rangle \mid (P, P)$	pattern
op		= += -=	update/access operator
t	::=	$v \mid t_1 \cdot \ldots \cdot t_n \mid \{x\} \mid \{x : x f t\}$	term
E		$0 \mid L \mid \langle S \rangle_{\!\sigma} \mid \sigma \rightsquigarrow \sigma' \mid (E \mid E)$	ecosystem

An LSA A includes the unique identifier ι , type tag, and a set of property/value associations $p_1=v_1, \ldots, p_n=v_n$ (p_i are not repeated). A value v can be an identifier λ (modelling a bond towards its LSA), primitive data value d or a sequence $v_1 \cdot \ldots \cdot v_n$. An LSA-space S is a composition of LSAs by operator " | ".

An eco-law L is a chemical-resembling reaction transforming a reagent set R into R' by speed r; a reagent set is a sum of LSA "patterns" *P*. One such pattern can be void (0), can specify the identifier of an LSA (ι) , its tag (type), or that property *p* either holds term t $(\langle p=t \rangle)$ or should be added/removed with a value (+= and -=). A term *t* is a value, possibly including some unconstrained variable $\{x\}$, or an *annotated variable* $\{x : x f t\}$ that should (semantically) match *t* by function *f*—such functions *f*, which are typically application-dependent, are binary fuzzy predicates over values, yielding a real number in between 0 (no match) and 1 (full match). Finally, the whole ecosystem *E* is a multiset composition of eco-laws *L*, LSA-spaces $\langle S \rangle_{\sigma}$ (σ is the space identifier), and space connections $\sigma \rightsquigarrow \sigma'$.

For the sake of space we do not formally model wellformedness of eco-laws, which dictate the possibility of using remote identifiers – as well as operators "+=" and "-=" – only in the right hand side; additionally, each pattern can specify at most one occurrence of an LSA identifier, a type, and a property.

Auxiliary functions We introduce a congruence relation " \equiv ", to equip the above syntax with additional features. We first establish equivalence of the following notations, which will be useful to match an LSA with a pattern:

$$\iota: type \langle p_1 = v_1, \dots, p_n = v_n \rangle \equiv \iota, type, \langle p_1 = v_1 \rangle, \dots, \langle p_n = v_n \rangle$$

We introduce equations giving semantics to an update function $R \triangleleft A$, which takes the right-hand side R of an eco-law and an LSA A to which the eco-law has to be applied, and enacts all updates specified in R which pertains A:

$$\begin{array}{c} (P+R) \blacktriangleleft A \equiv (P \blacktriangleleft A) + (R \blacktriangleleft A) & 0 \blacktriangleleft A \equiv 0 \\ (\iota,P) \bigstar (\iota,P') \equiv \iota, (P' \triangleleft P) & (\iota,P) \blacklozenge (\iota',P') \equiv (\iota,P) \text{ if } \iota \neq \iota' \\ P \triangleleft (P',P'') \equiv (P \triangleleft P') \triangleleft P'' & 0 \triangleleft P \equiv P \\ P, \langle p = \upsilon \rangle \triangleleft p = \upsilon' \equiv P, \langle p = \upsilon' \rangle & P, type' \triangleleft type \equiv P, type \\ P, \langle p = \upsilon \rangle \triangleleft \langle p + = \upsilon' \rangle \equiv P, \langle p = \upsilon \cdot \upsilon' \rangle \\ P, \langle p = \upsilon \lor \lor \lor \land \langle p - = \upsilon' \rangle \equiv P, \langle p = \upsilon \lor \upsilon' \rangle \\ P, \langle p = \upsilon \lor \lor \lor \lor \triangleleft \langle p - = \upsilon' \rangle \equiv P, \langle p = \upsilon \rangle \\ P, \langle p = \upsilon \lor \triangleleft \langle p' \circ \upsilon' \rangle \equiv (P \triangleleft \langle p' \circ \upsilon' \rangle), \langle p = \upsilon \rangle \text{ if } p \neq p' \end{array}$$

Namely, operator " \blacktriangleleft " affects the left-hand side only if it specifies the same LSA as the right-hand side, in which case it proceeds by updating through operator \triangleleft , which in turn changes (replaces, adds or removes) values of all properties of the LSA. For instance (for the case of a single-reagent R), we have:

$$\begin{array}{l} (\lambda, \texttt{type}, \langle\texttt{c=5}\rangle, \langle\texttt{a+=3}\rangle) \blacktriangleleft \lambda \texttt{:}\texttt{type} \langle\texttt{a=5}, \texttt{b=4}\rangle \equiv \\ \lambda \texttt{:}\texttt{type} \langle\texttt{a=5} \cdot 3, \texttt{b=4}, \texttt{c=5}\rangle \end{array}$$

Finally, we consider " \equiv " as the largest relation which, other than equations above, also handles operators "+", " |", ",", and "." as multiset ones (i.e, commutative, associative, and absorbing 0).

Matching Given any syntactic structure s (an eco-law L, an LSA A, and so on), we use standard notation $[v_1/x_1, \ldots, v_n/x_n]s$ for s after substituting variables x_1, \ldots, x_n with values v_1, \ldots, v_n —this is syntactic as usual, i.e., $[v/x]\{x\} = [v/x]\{x : x f v'\} = v$. We now need a mechanism to match the partial specification of a pattern P with the complete one of an LSA A: we hence abuse the notation writing [A/P] for the most general substitution (if any exists) which applied to pattern (A, P) equates it to A (note this makes sense only if properties in A and P do not clash); for instance:

$$[\lambda: \texttt{type} \langle \texttt{n=lit}, \texttt{s=40}, \texttt{u} = \lambda_1 \cdot \lambda_2 \rangle / (\{i\}, \texttt{type}, \langle \texttt{u} = \lambda_1 \cdot \{u\} \rangle, \langle \texttt{s=}\{s\} \rangle)]$$

yields substitution $[\lambda/i, 40/s, \lambda_2/u]$.

Finally, following and refining the approach in [21], we introduce a mechanism of fuzzy/semantic matching, by which we rate the extent to which an LSA matches a patter. Function $\mu(A, P) \in [0, 1]$ is introduced to this end—note this will be actually used only if notation [A/P] actually yields a substitution, which simplifies definition below. This function is defined as follows:

$$\mu((id, type, P), (id', type', P')) = \mu(P, P') \mu((\langle p=v \rangle, P), (\langle p=\{x : x f v'\} \rangle, P')) = \mu(P, P') * f(v, v') \mu(P, P') = 1 \text{ otherwise}$$

So, considering the application-dependent function matches, we have e.g.:

 $\begin{array}{ll} \mu(& \lambda: \texttt{type} \langle \texttt{n=basket}, \texttt{s=40} \rangle, \\ & \lambda: \texttt{type} \langle \texttt{n=} \{ x: x \text{ matches sport} \}, \texttt{s=40} \rangle &) = \\ & \texttt{matches}(\texttt{basket}, \texttt{sport}) \end{array}$

Operational semantics The operational semantics of this calculus is given as a CTMC (Continuous-time Markov Chains) model, like other stochastic calculi for chemical-like behaviour [21]. A transition system $(\mathbb{E}, \rightarrow, \mathbb{R}_0^+)$ is defined where transitions are of the kind $E \xrightarrow{r} E'$, meaning that ecosystem in state $E \in \mathbb{E}$ moves to $E' \in \mathbb{E}$, by either an eco-law reaction or by an LSA diffusion, with Markovian rate $r \in \mathbb{R}_0^+$ —namely the transition duration is a stochastic variable following negative exponential distribution with average value 1/r time units.

The transition relation is defined by the rules in Figure 2. Rule (C) and (P) provide standard semantics of congruence and parallel composition: the former states that transitions are to be applied modulo congruence " \equiv "; the latter that any subsequent rule is actually local, since any ecosystem subpart E' is allowed to move to a E'' in isolation. Rule (Ef) handles the special case of an eco-law with no reagents, in which case simply the right-hand side R of the eco-law is formed by LSAs to be inserted in any space σ with rate r. Rule (Re) is recursive, and uses (Ef) when termination is

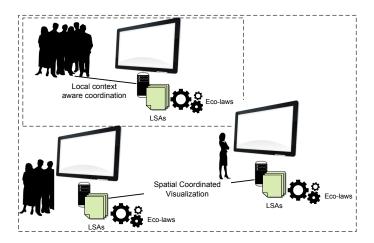


Figure 3. (top). Local Context-Aware Coordination A public display collects and merges together detailed information about several user profiles. bottom). Spatial Coordinated Visualisations Public displays interact to present coherent coordinated content (e.g., to provide steering and directions) to users.

reached: it takes one pattern P in the left-side of the ecolaw, and tries to match it with some LSA A in space σ . The resulting state S' in space σ is computed by recursively applying (Re) to a space without A, without P in the ecolaw, by applying substitution [A/P] to the remainder of the eco-law, and finally updating the right-hand side with LSA A. Finally, rule (Df) is used to ship (at an arbitrary high rate) some remote LSA with id λ , tag type, and properties P, from space σ into a neighbouring space σ' .

IV. USE CASES

Displays providing information are one of the first technologies that is truly becoming pervasive and ubiquitous. Public displays are gradually deployed everywhere: from airports and stations to city centres. At the same time, an increasing number of people carry on private displays embedded in their smart phones. These devices are gradually provided with sensing and networking capabilities to perceive their environment and act accordingly. Since displays can (and should) interact with each other the resulting scenario is that of an ecosystem of displays more deeply fulfilling users' information needs. Given the massive amount of devices comprising such an ecosystem, a natural coordination problem arises: how can displays coordinate their activities to optimise the user experience? How can displays acquire data from sensors and elaborate them so as to get an overall better representation of the context? How can displays coordinate so as to steer a crowd in a spatial system? We here discuss some case study involving different coordination patterns to be enacted by the display ecosystem in the form of suitable eco-laws.

(C)

$$\frac{E \equiv E_{1} \qquad E_{1} - \frac{r}{r} \rightarrow E'_{1} \qquad E'_{1} \equiv E'}{E - \frac{r}{r} \rightarrow E'}$$
(P)

$$\frac{E' - \frac{r}{r} \rightarrow E'}{E \mid E' - \frac{r}{r} \rightarrow E \mid E''}$$
(Ef)

$$\frac{-}{(0 \mapsto_{r} R) \mid \langle S \rangle_{\sigma} - \frac{r}{r} \rightarrow (0 \mapsto_{r} R) \mid \langle S \mid R \rangle_{\sigma}}{(P + R \mapsto_{r} R') \mid \langle A \mid S \rangle_{\sigma} - \frac{r' + \mu(A, P)}{r} (P + R \mapsto_{r} R') \mid \langle S' \rangle_{\sigma}}$$
(Re)

$$\frac{(R[A/P] \mapsto_{r} R'[A/P] \blacktriangleleft A) \mid \langle S \rangle_{\sigma} - \frac{r'}{r} \rightarrow L \mid \langle S' \rangle_{\sigma}}{(P + R \mapsto_{r} R') \mid \langle A \mid S \rangle_{\sigma} - \frac{r' + \mu(A, P)}{r} (P + R \mapsto_{r} R') \mid \langle S' \rangle_{\sigma}}$$
(Df)

$$\frac{-}{(P + R \mapsto_{r} R') \mid \langle A \mid S \rangle_{\sigma} - \frac{r' + \mu(A, P)}{r} (P + R \mapsto_{r} R') \mid \langle S' \rangle_{\sigma}}$$

 $\langle (\lambda^{\leadsto}, type, P) \mid S \rangle_{\!\sigma} \mid \sigma \rightsquigarrow \sigma' \mid \langle S' \rangle_{\!\sigma'} \xrightarrow{\quad \infty \quad} \langle S \rangle_{\!\sigma} \mid \sigma \rightsquigarrow \sigma' \mid \langle (\lambda, type, P) \mid S' \rangle_{\!\sigma'}$

Figure 2. Operational Semantics of Eco-laws

A. Local Context-Aware Coordination

The eco-law framework can support public displays in retrieving data from different kinds of sensor and to use them to infer high-level context information, either automatically or through some external recommendation system.

Let's consider the case of a public area (a mall, an airport) where some people wander around with their portable device (e.g. a smartphone) that keep their profile. Additionally, such user 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, the accelerometer if the user is standing or walking, and so on). A number of public displays are deployed to show different kinds of visualisation services to users, e.g. monitoring, advertisements, directions (i.e. crowd steering). Public displays have to acquire data about the users standing in front of them and get a synthetic representation of what is happening in order to provide the best possible content to users. Also public displays have an associated LSA that specifies the display characteristics and enables interactions with users' in proximity. A key role is played by the SAPERE nodes hosting LSA-spaces: they are spread in the environment, and as soon as they sense a device or service around they reify their presence by a proper LSA. In particular, spaces will hosts LSAs for the public displays, the profile and sensors of users around, and the visualisation services available in that context (see Figure 3-top). Eco-laws handling this scenario (filling the context of each public display, and accordingly choosing the most proper visualisation service as a summary of the current audience), are shown in Figure 4-we use a concrete syntax that almost fully adhere to the abstract one described in previous section.

Eco-law [CTX-SNS] states that a display's LSA d with property contextualising set to true gets automatically bonded to any sensor's LSA s in the same space, namely, s gets added to the display's context property this happens if s is not already there, in which case this reaction is not fired for it has no neat effect. Similarly, eco-laws [CTX-USR] and [CTX-SER] bond the display's LSA to that of any user around and of any available visualisation service: in particular, a service is added only if the required screen properties p' match with the available screen properties p—matches function is here in charge of (possibly semantically) match display property lists, facilitating connection with compatible services.

A decision about which service is to be visualised at a given time is made by updating property showFrom in the display's LSA. The display can achieve this in two ways. On the one hand, it monitors its LSA, reasons about context property (it can access LSAs of sensors, users, and service content by navigating LSA references through its LSA), takes a choice about what service to visualise, and accordingly reifies it in showFrom property. On the other hand, the display can externalise the decision by setting recommend property to ask. By eco-law [SEL-ASK] a recommendation LSA is then created with all the necessary information to take a meaningful decision. One possibility for getting a reply is by eco-law [SEL-MATCH], which finds a bonded service s whose content matches the display's context, and accordingly update answer property in the recommendation-the semantics of eco-laws make sure that the most proper choice is probabilistically taken. Alternatively, and mostly transparently, as the recommendation LSA has been created an available recommender agent (which can possibly encapsulate some situation recognition algorithm) – a pervasive service eventually injected in the system to improve the quality of visualisation - can intercept such LSA and provide itself a reply.

As a decision is taken, eco-law [ACT] starts display/service interaction, by creating a communication channel between the display's LSA (property channel) and a new LSA of type channelProxy, acting as a reification of the interaction session—channels have not been modelled in previous section for the sake of space, and for this is a quite orthogonal mechanism. Additionally, by eco-law [LOG] one or more log LSAs are created to store the historical data related to the present event, namely, the display d visualising the selected service s to the user u. The above eco-laws should come with proper rates, to be tuned according to the needs of the application at hand.

```
{d}:display[contextualizing = true] + {s}:sensor
 ->[CTX-SNS]
{d}:[context += {s}] + {s}
{d}:display[contextualizing = true] + {u}:user
 ->[CTX-USR]
{d}:[context += {u}] + {u}
{d}:display[contextualizing = true, screenprops = {p}] +
{s}:service[screenprops = {p': p' matches p}]
 ->[CTX-SER]
{d}:[context += {s}] + {s}
{d}:display[recommend = ask, context = {ctx}]
 ->[SEL-ASK]
{d}:display[recommend = asked] +
{r}:recommenation[device={d}, context={ctx}, question=service]
{r}:recommenation[device={d}, context={ctx}, question=service] +
{s}:service[content={c: c matches ctx}]
 ->[SEL-MATCH]
\{s\} + \{r\}: [answer=\{s\}]
{d}:display[recommend = asked, context = {ctx}] + {r}:recommenation[device={d}, answer={s}]
->[SEL-REP]
{d}:[recommend = received, showFrom = {s}]
{d}:display[showFrom={s},channel=null]
-->[ACT]
{d}:[channel=#sink] + {c}:channelProxy[service={s},display={d},channel=#source]
{d}:display[showFrom={s},context has {u}] + {u}:user
-->[LOG]
{d} + {u} + {l}:log[time=#current-time,user={u},service={s},display={d}]
```

Figure 4. Eco-laws for Local Context-Aware Coordination

B. Spatial Coordinated Visualisation

The idea of crowd-steering is to guide (steer) a group of users in moving through an environment by providing visual cues by means of public and private displays. In this use case, we focus on public displays coordinating so as to direct users to a near display that is actually providing a content that best matches the user's interest (see Figure 3bottom). This scenario involves several interactions patterns: (*i*) public displays diffuse information about the content they are actually visualising; (*ii*) by such diffused information, and by users in front of the screen, displays acquire global context information; (*iii*) the infrastructure, by interaction with the users' private displays, can provide them with coherent information to steer them to the most appropriate display. Such behaviour can be enacted by the eco-laws in Figure 5.

Eco-law [PUMP] is used to initiate the generation of a computational field [12], namely, a distributed structure of LSAs advertising the existence and state of a display into a whole part of the network. In particular, as a display has the diffuse property set to true, a field LSA is locally created. Eco-law [DIFF] is used to make any field LSA spawn a remote one, with distance property updated by adding variable #dist—this will be bound to the distance of the neighbouring node as soon as this LSA is shipped there. Note this eco-law activates only if distance is smaller than the expected range of the field—

function greater-than yields 1 if left-hand argument is greater than right-hand one, 0 otherwise, and similarly for is-sum-of. Eco-law [MIN] completes the creation of the field, choosing in each LSA-space the field LSA with smaller distance from the field source. Ultimately, the above three eco-laws reify an LSA in each space within range distance, carrying a numerical value holding the minimum distance from the source display—which can hence be used to also identify a minimum path towards it [12].

We assume now we are in the situation in which all displays are contextualised by eco-laws [CTX-SNS,CTX-USR,CTX-SER], and diffused a computational field by [PUMP,DIFF,MIN], by which the infrastructure can steer users: this is achieved by eco-laws [STE-ASK,STE-MATCH, STE-REP], which are structurally similar to [SEL-ASK, SEL-MATCH, SEL-REP]. Eco-law [STE-ASK] intercepts the situation in which a user u in front of a displays has a profile p that does not match currently visualised content s: in this case, a new recommendation LSA is created, asking for a proper display for u. Either a recommender or eco-law [STE-MATCH] solve this situation; the latter looks for the computational field f of a display visualising content c that is compatible with u's profile, and accordingly updates recommendation LSA by proposing display d-as usual, the semantics of eco-laws make sure that the most proper choice is probabilistically taken. Then, eco-law [STE-REP] incorporates this decision by updating the user's LSA, i.e., updating property suggestion shall make the user's

```
{d}:display[diffuse=true,range={r},showFrom{s}] + {s}:service[content={c}]
 ->[PUMP]
{d} + {s} + {f}:field[display={d}, content={c}, range={r}, distance=0]
{f}:field[display={d},content={c},distance={r'},range={r:r greater-than r'}]
 ->[DIFF]
remote {f'}:field[display={d},content={c},distance={mt': mt' is-sum-of mt,#dist},range={r}]
{f}:field[display={d},distance={mt}] +
{f'}:field[display={d}, distance={mt': mt' greater-than mt}]
 ->[MIN]
{f}
{d}:display[showFrom={s}, context = {ctx}, context has {u}] +
{s}:service[content={c: c not-matches p}] + {u}:user[profile={p}]
 ->[STE-ASK]
{d} + {s} + {u} + {r}:recommendation[subject={u}, context={ctx}, question=display]
{r}:recommendation[subject={u}, question=display]
{u}:user[profile={p}] + {f}:field[source={d},type={display},content={c: c matches p}]
 ->[STE-MATCH]
{f} + {u} + {r}:recommendation[answer={d}]
{u}:user + {r}:recommendation[subject={u}, answer={d}]
--> [STE-REP]
{u}:user[suggestion={d}]
```

Figure 5. Eco-laws for Coordinated Visualisation

PDA signal the recommendation of moving to a new area, where the suggested display is located.

V. RELATED WORK

Chemical-oriented coordination The issue we face in this article can be framed as the problem of finding the proper coordination model for enabling and ruling interactions of pervasive services. We take as ground the archetypal LINDA model, which simply provides for a blackboard with associative matching for mediating component interactions through insertion/retrieval of tuples. Then, we followed the idea of engineering the coordination space of a distributed system by some policy "inside" tuple spaces, following the pioneer works of approaches like TuCSoN [15] and MARS [5]. In particular, our proposal tries to extend these models to include bio-inspired ecological mechanisms, by fine-grained and well structured chemical-like reactions. In particular, the coordination approach we propose in this paper originates from the chemical tuple space model in [21], though with some notable differences: (i) here we provide a detail notational framework to flexibly express eco-laws that work on patterns of LSAs and affect their properties; (ii) the chemical concentration mechanisms proposed in [21] to exactly mimick chemistry is not mandatory here-though it can be achieved by a suitable design of rate expressions; *(iii)* the way we conceive the overall infrastructure, and relationship between agents and their LSAs goes beyond the mere definition of the tuple-space model.

Chemistry has been a source of inspiration for several works in (distributed) computing and coordination like in the Gamma language and its extensions [3]. The main features we inherit from this research thread include: (i) conferring a high-level, abstract, and nature-inspired

character to the language used to program the distributed system behaviour; *(ii)* providing a reactive computational model very useful in autonomic contexts. While Gamma and it extensions (such as HOCL) were exploited in different application contexts [3], they originated with the goal of writing concurrent, general-purpose programming languages. Our approach instead aims at specifically tackling coordination infrastructures for pervasive systems, which calls for dictating specific mechanisms of fuzzy matching, diffusion, context- and spatial-awareness, and agent-LSA interaction.

Situatedness and Context-Awareness Considering the issues of situatedness and context-awareness, extensions or modifications to the traditional SOAs have been recently proposed to address adaptivity in pervasive environments. Similarly to our approach, in PLASTIC [1] service descriptions are coupled with dynamic annotations related to the current context and state of a service, to be used for enforcing adaptable forms of service discovery. However, our approach gets rid of traditional discovery services and enforces dynamic and adaptive service interaction via simple chemical reactions and a minimal middleware.

In many proposals for pervasive computing environments and middleware infrastructures, the idea of "situatedness" has been promoted by the adoption of shared virtual spaces for services and components interactions. The pioneering system Gaia [18] introduces the concept of active spaces, that is active blackboard spaces acting as the means for service interactions. Later on, a number of Gaia extensions where proposed to enforce dynamic semantic pattern-matching for service composition and discovery [7] or access to contextual information [6]. Other related approaches include: Egospaces [9], LIME [13] and TOTA [12]. Our model shares the idea of conceiving components as "living" and interacting in a shared spatial substrate (of tuple spaces) where they can automatically discover and interact with one another. Yet, our aim is broader, namely, to dynamically and systemically enforce situatedness, service interaction and data management with a simple language of chemical reactions, and most importantly, enacting an ecological behaviour.

Self-organisation Several recent works exploit the lessons of adaptive self-organising natural and social systems to enforce self-awareness, self-adaptivity and self-management features in pervasive computing systems. At the level of individual component modelling, these proposals take the form of either situated reactive agents or proactive and goal-oriented ones [17]. At the level of interaction models, these proposals typically take the form of specific natureand socially-inspired interaction mechanisms [2], enforced either at the level of component modelling or via specific middleware-level mechanisms.

We believe our framework integrates and improves these works in three main directions: (i) it abstracts from the specific internal characteristics of components (no matter whether they are simple reactive components or complex goal-oriented ones) and rather proposes an approach that seamlessly applies to both cases; (ii) it tries to identify an interaction model that is able to represent and subsume the diverse nature-inspired mechanisms under a unifying selfadaptive abstraction (i.e. the semantics chemical reactions); (iii) the ecological approach we undertake goes beyond most of the current studies that limit to ensembles of homogeneous components, defining a suitable framework for supporting the vision of novel pervasive and Internet scenarios as made up of self-adaptive devices and services, that autonomously cooperate for the creation of global services.

VI. CONCLUSION AND FUTURE WORK

In this paper we outlined the SAPERE architecture, focussing on the chemical-resembling eco-laws framework for the self-adaptive, context-aware coordination of pervasive computing systems—although we focussed on the ecosystem of display scenario, the proposed model and language could be applicable also to other coordination domains, including advanced adaptive and real-time traffic control, and social and augmented reality services. The road towards the full realisation of the SAPERE framework is necessarily longer than the scope of this paper, and includes deeper evaluation of the proposed model, design of an efficient and lightweight tuple-space infrastructure supporting LSAs and eco-laws, and implementation of use cases based on the most recent pervasive computing technologies. Concerning the model described in this paper, current and future works focus on: (*i*) relying on standard technologies like RDF for the actual incarnation of LSAs and eco-laws, (*ii*) the use of OWL ontologies for promoting the conception of an application-independent set of eco-laws, (*iii*) the adoption of more advanced spatial mechanisms into eco-laws, to support higher-level self-organisation structures of LSAs.

Acknowldgments This work has been supported by the EU-FP7-FET Proactive project SAPERE – Self-aware Pervasive Service Ecosystems – under contract no. 256874.

REFERENCES

- M. Autili, P. Benedetto, and P. Inverardi. Context-aware adaptive services: The plastic approach. In *FASE '09: Proceedings of the 12th International Conference on Fundamental Approaches to Software Engineering*, pages 124–139, Berlin, Heidelberg, 2009. Springer-Verlag.
- [2] O. Babaoglu, G. Canright, A. Deutsch, G. A. D. Caro, F. Ducatelle, L. M. Gambardella, N. Ganguly, M. Jelasity, R. Montemanni, A. Montresor, and T. Urnes. Design patterns from biology for distributed computing. *ACM Trans. Auton. Adapt. Syst.*, 1(1):26–66, 2006.
- [3] J.-P. Banâtre and T. Priol. Chemical programming of future service-oriented architectures. *Journal of Software*, 4:738– 746, September 2009.
- [4] N. Busi, R. Gorrieri, and G. Zavattaro. On the expressiveness of linda coordination primitives. *Inf. Comput.*, 156(1-2):90– 121, 2000.
- [5] G. Cabri, L. Leonardi, and F. Zambonelli. MARS: A programmable coordination architecture for mobile agents. *IEEE Internet Computing*, 4(4):26–35, 2000.
- [6] P. D. Costa, G. Guizzardi, J. P. A. Almeida, L. F. Pires, and M. van Sinderen. Situations in conceptual modeling of context. In *Tenth IEEE International Enterprise Distributed Object Computing Conference (EDOC 2006), 16-20 October* 2006, Hong Kong, China, Workshops, page 6. IEEE Computer Society, 2006.
- [7] C.-L. Fok, G.-C. Roman, and C. Lu. Enhanced coordination in sensor networks through flexible service provisioning. In J. Field and V. T. Vasconcelos, editors, *Coordination Languages and Models*, volume 5521 of *LNCS*, pages 66– 85. Springer-Verlag, June 2009. 11th International Conference (COORDINATION 2009), Lisbon, Portugal, June 2009. Proceedings.
- [8] M. N. Huhns and M. P. Singh. Service-oriented computing: Key concepts and principles. *IEEE Internet Computing*, 9(1):75–81, 2005.
- [9] C. Julien and G.-C. Roman. Egospaces: Facilitating rapid development of context-aware mobile applications. *IEEE Trans. Software Eng.*, 32(5):281–298, 2006.
- [10] M. P. Locatelli, M. Loregian, and G. Vizzari. Artificial Societies in a Community-Based Approach to Ambient Intelligence. *The Computer Journal*, 53(8):1152–1168, 2010.

- [11] M. Mamei, R. Menezes, R. Tolksdorf, and F. Zambonelli. Case studies for self-organization in computer science. *Journal of Systems Architecture*, 52(8):443–460, 2006.
- [12] M. Mamei and F. Zambonelli. Programming pervasive and mobile computing applications: The tota approach. ACM Trans. Softw. Eng. Methodol., 18(4):1–56, 2009.
- [13] A. L. Murphy, G. P. Picco, and G.-C. Roman. Lime: A coordination model and middleware supporting mobility of hosts and agents. ACM Trans. on Software Engineering and Methodology, 15(3):279–328, 2006.
- [14] R. D. Nicola, D. Latella, and M. Massink. Formal modeling and quantitative analysis of KLAIM-based mobile systems. In SAC '05: Proceedings of the 2005 ACM symposium on Applied computing, pages 428–435, New York, NY, USA, 2005. ACM Press.
- [15] A. Omicini and F. Zambonelli. Coordination for Internet application development. *Autonomous Agents and Multi-Agent Systems*, 2(3):251–269, Sept. 1999.
- [16] J. pierre Banâtre, P. Fradet, and Y. Radenac. Classical Coordination Mechanisms in the Chemical Model. 2009.
- [17] A. Ricci, A. Omicini, M. Viroli, L. Gardelli, and E. Oliva. Cognitive stigmergy: Towards a framework based on agents and artifacts. In D. Weyns, H. V. D. Parunak, and F. Michel, editors, *Environments for MultiAgent Systems*, volume 4389 of *LNAI*, pages 124–140. Springer, Feb. 2007. 3rd International Workshop (E4MAS 2006), Hakodate, Japan, 8 May 2006. Selected Revised and Invited Papers.
- [18] M. Román, C. K. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt. Gaia: a middleware platform for active spaces. *Mobile Computing and Communications Review*, 6(4):65–67, 2002.
- [19] P. V. Roy, S. Haridi, A. Reinefeld, J.-B. Stefany, R. Yap, and T. Coupaye. Self-management for large-scale distributed systems: an overview of the selfman project. In *Formal Methods for Components and Objects, LNCS No. 5382*, pages 153–178. Springer Verlag, 2008.
- [20] M. Ulieru and S. Grobbelaar. Engineering industrial ecosystems in a networked world. In 5th IEEE International Conference on Industrial Informatics, pages 1–7. IEEE Press, 2007.
- [21] M. Viroli and M. Casadei. Biochemical tuple spaces for self-organising coordination. In *Coordination Languages and Models*, volume 5521 of *LNCS*, pages 143–162. Springer-Verlag, June 2009.
- [22] M. Viroli and F. Zambonelli. A biochemical approach to adaptive service ecosystems. *Information Sciences*, 180(10):1876–1892, 2010.