Architecture and Metaphors for Eternally Adaptive Service Ecosystems

Franco Zambonelli[1] and Mirko Viroli[2]

[1] DISMI – Universita' di Modena e Reggio Emilia
42100 Reggio Emilia, Italy
franco.zambonelli@unimore.it

[2] DEIS – University of Bologna
47023 Cesena (FC), Italy
mirko.viroli@unibo.it

Summary. In this paper, we first motivate the need for innovative open service frameworks that ensure capability of self-adaptability and long-lasting evolvability (i.e., eternity). On this basis, we discuss how such frameworks should get inspiration from natural ecosystems, by enabling modelling and deployment of services as autonomous individuals in an ecosystem of other services, data sources, and pervasive devices. A reference architecture is presented to clarify the concepts expressed, and then several possible approaches to realise the idea are surveyed and critically analyzed.

1 Motivations

In the near future, pervasive sensing and actuating devices will densely populate our everyday environments, will be tightly integrated with current Telecom and Internet networks, and will eventually contribute to blur the distinction between Telecom and Internet networks [5, 9].

In this context of tight convergence and integration, a single innovative open software platform will have to be provided to host and orchestrate in an integrated and self-managing way the execution of general-purpose pervasive Telecom/Web services and the organization of large masses of contextual data. Also, such an infrastructure should take into account the increasingly diverse and demanding needs of users (which will also seamlessly act as consumers and producers of data and services) [14], and must be able to flexibly tolerate evolutions over time without requiring significant re-engineering to incorporate innovations and changing needs.

Recently, a great deal of research activity has been devoted to produce solutions to match the emerging characteristics of future networks [7] and to solve problems related to, e.g., increasing dependability, reducing management efforts via self-* features, enforcing context-awareness and adaptability, tolerating evolution over time and eventually ensure that the overall service framework (if not all services within) can be highly adaptive and very long-lasting, even in the absence of explicit management actions. Unfortunately, most of the solutions so far are proposed in terms of "add-on", one-of solutions to be integrated in existing frameworks. The result of this process is often an increased complexity of current frameworks and the emergence of contrasting trade-off between different solutions.

For instance, while the strict layering of network architectures and protocols enables services to easily survive changes at the device and communication level, it prevents information about the current execution context of services from freely flowing in the network, limiting the adaptability of services. As another example, a huge amount of research proposes application-level and self-organizing overlay networks as a mechanism to exchange information in several distributed and dynamic scenarios. However, such overlays are typically conceived to serve specific classes of application problems in specific network scenarios, and cannot tolerate adaptations with regard to changes in their usage or in the characteristics of the underlying network.

In our opinion, there is need for tackling all the above problems by reformulating their foundation, and by trying to answer the following question: *Is it possible to conceive a radically new way of modelling integrated network services and their execution environments, such that the apparently diverse issues of enabling pervasiveness, context-awareness, dependability, openness, diversity, flexible and robust evolution, can all be uniformly addressed once and for all?* In other words, in our opinion, the way towards the realisation of eternally adaptive services is to tackle the problem form the foundation, and start from a total deconstruction of current service architectures and models. We should no longer see services as localised "loci" of data and functionalities, whose activities are to be orchestrated and synchronised according to specific patterns, with the support of middleware services such as discovery services, routing services, data and context services, and where self-adaptability and evolvability are enforced via the introduction of autonomic managers [11]. Rather, we should start taking inspiration from natural systems [16, 8], where adaptability and eternal evolvability are there because of the basic "rules of the game".

No matter whether one thinks at physical systems, at chemical systems, at biological systems, as well as at general ecological systems. In all these systems, you can always recognise the following characteristics: Above a common environmental substrate (defining the basic "laws of nature" and the ground on which individuals can live), individuals of different kinds (or species) interact, compete, and combine with each other (in respect of the basic laws of nature), so as to serve their own individual needs as well as the sustainability and the evolvability of the overall system.

Although such considerations apply whether you think at a physical system, at chemical systems, or at biological system, let us try to better elaborate the idea by referring at biological evolution and at the dynamics of life on earth. Life is, to most extents (i.e., apart from planetary catastrophes) sort of eternal and eternally adaptive: mechanisms, protocols, and the basic "infrastructure" of life do not change and have never changed since its first appearance. Simply, new individuals get on appearing, finding their own way in the overall system, and possibly leading to the emergence of new reactions and new combinations of elements, in the end possibly leading to the emergence of new life forms and new ecological dynamics. The chemistry of life is eternal, the forms under which it manifests depends on the specific characteristics of the environment, and the specific contingencies occurring in the environment, and on the specific species that populate the environment. Also, very important, it is not individuals that evolve, but the ecosystem as a whole.

This is the sort of endeavour that we think one should assume towards the realisation of "Eternally adaptive service ecosystems": conceiving services and data components as individuals in an open ecosystem in which they interact accordingly to a limited set of "eco-laws" to serve their own individual purposes in respect of such laws, and where self-adaptation and eternity are inherent, endogenous properties of the ecosystem rather than peculiar characteristics of its individuals [10].

Against this background, the remainder of this paper *(i)* tries to sketch a general reference architecture for nature-inspired service ecosystems and *(ii)* surveys and shortly analyses the possible metaphors that can be adopted for such ecosystems, and that can lead to different realisation of the reference architecture.

2 A Reference Architecture for Eternally Adaptive Service Ecosystems

Independently of the specific approach adopted, a uniform reference architecture can be adopted for open service ecosystems. A pictorial representation of such an architecture is reported in Figure 1.

At the very low level, the physical ground on which the ecosystem will be deployed is laid, which is a very dense network (ideally, a pervasive continuum) of networked computing devices and information sources. The former includes all the devices that are going to increasingly pervade all our everyday environments (e.g., PDAs, smart phones, sensors, tags), all interconnected with each other. The latter includes the increasing amount of Web data sources that already (and increasingly) collect knowledge, facts and events about nearly every aspect of the world.

At the top level, service developers, producers and consumers of services and data, access the open service framework for using/consuming data or services, as well as for producing and deploying in the framework new services and new data components. At both the lower and the top levels of the architecture openness stands: on the one hand, new devices can join/leave the system at any time; on the other hand, new users can interact with the framework and can deploy new services and data items on it. Between these two levels, the components of the ecosystem reference architecture stand.

The level of "Species" is the one in which physical and virtual devices of the pervasive system, digital and network resources of any kind, persistent and temporary knowledge/data, contextual information, events and information requests, and of course software service components, are all provided with a uniform abstract view of being the"living entities" of the system, which we refer to as *ecosystem individuals*, and which populate the world. Although such individuals are expected to be modelled (and computationally rendered) in a uniform way, they will have specific characteristics very different from each other, i.e., they will be of different “species”.

In a bootstrap phase, an ecosystem is expected to be filled with a set of individuals physically deployed in the environment (physical and network resources, contextual information, initialization data and services, and so on). From then on, the ecosystem eternally lives, with the population of individuals evolving in different ways: *(i)* the initial set of individuals is subject to changes (to tackle the physical system’s mobility, faults, and evolution); *(ii)* service developers and producers inject in the system new individuals (developers insert new services and virtual devices, prosumers insert data and knowledge); and *(iii)* consumers keep observing the environment for certain individuals (inject information requests and look for certain data, knowledge, and events).

Below the level of species there is the individuals “world” level, which provides the virtual fabric that supports individuals, their activities and interactions, as well as their insertion and evolution. This overall ecological behaviour is to be enacted by a middleware substrate, a software infrastructure deployed on top of the physical deployment context (i.e., on top of the pervasive continuum), which is in charge of handling insertion and observation of individuals, their persistence and accessibility, as well as their interaction. Moreover, it should deal with the mobility and dynamism of the underlying context, properly turning it into the creation, destruction and change of individuals representing physical and network resources. More in general. the world level define the “shape” and the characteristics of the world in which individuals live.

The way in which individuals interact, compose with others, aggregate so as to form or spawn new individuals, and decay (ultimately winning or losing the natural selection process intrinsic in the ecosystem) is determined by the set of fundamental “laws” regulating the eternal service ecosystems model. Starting from the unified description of living entities—the information/service/structure they provide—and from proper matching criteria, such laws basically specify the likelihood of certain spontaneous evolutions of individuals or groups of individuals. Typical evolution patterns driven by such laws are the following: temporary data and services decay as long as they are not exploited until disappearing, and dually, they get reinforced when exploited; data, data requests, and data retrieving services might altogether match, hence spawning “data-found” events; new services can be created by aggregating existing services whose description strongly matches; an existing service can be upgraded—or corrected when faulty—by injecting a service patch that will automatically compose to it; and so on. A key consequence of the fact that all components are seamlessly seen as individuals is that ecological laws abstract away from the peculiarities of the above cases, uniformly dealing with the concepts of individuals’ match-based grouping and evolution: laws of the ecology are the only part of the ecosystem that do not evolve—as happens in natural ecologies. Accordingly, services and applications built with this paradigm will never be stopped or shutdown, but simply debugged, maintained, sustained, or sacrificed by an on-the-fly evolution based on laws behaviour and on the insertion of new individuals—e.g., acting like cures, diseases, feeding resources or viruses.

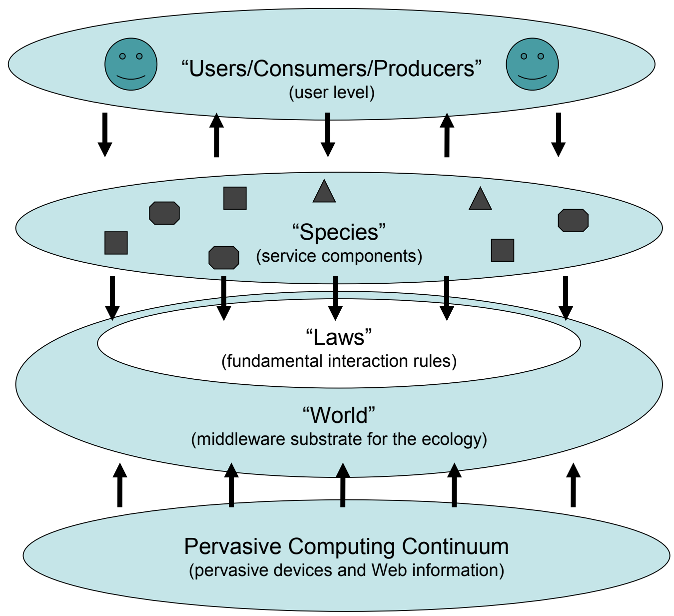

Fig. 1. Service Ecosystems Architecture

3 Survey and Analysis of Possible Approaches

The key difference in the possible approaches that can be undertaken towards the realisation of eco-inspired service frameworks (as from the described reference architecture) stands in the metaphor adopted to model the ecosystem, its individuals, and its laws. In particular—without excluding the existence of other useful natural metaphors or the possibility of conceiving interesting non-natural metaphors—the main metaphors that can be adopted and have been suggested are: physical metaphors [6, 12], chemical metaphors [3], biological metaphors [2, 4, 15], together with the properly called ecological metaphors [1, 13].

A summary of the characteristics of each of these metaphors is in Figure 2. We emphasise in any case that, so far, none of these metaphors has been actually adopted to extensively study and prototype an actual, open and general-purpose service framework: either the metaphor has been applied to specific application scenarios [12, 4, 15] or its potential general adoption has been only envisioned [6, 1].

3.1 Metaphors

Let us now come to the distinguishing characteristics of each metaphor, a summary of which is in Figure 2.

Physical metaphors consider that the species of the ecosystem are sort of computational particles, living in a world of other particles and virtual computational fields, which act as the basic interaction means. In fact, all activities of particles are driven by laws that determine how particles should be influenced by the local gradients and shape of the computational field: they can change their status based on the perceived fields, and they can move or exchange data by navigating over such fields (i.e., by having particles that move following the gradient descent of a field, or by making them spread sort of data particles to be routed according to the shape of fields). The world in which such particles live and in which fields spread and diffuse can be either a simple (euclidean) metric world, or it could be a sort of relativistic world, in which shapes and distances in the environment are not "inherent" but are rather shaped by fields themselves (as in gravitational space-time).

Chemical metaphors consider that the species of the ecosystem are sorts of computational atoms/molecules, with properties described by some sort of semantic descriptions which are the computational counterpart of the description of the bonding properties of physical atoms and molecules. Indeed, the laws that drive the overall behaviour of the ecosystem are sort of chemical laws, that dictates how chemical reactions and bonding between components take place (i.e., relying on some forms of pattern matching between the semantic description of components), and that can lead to both the production of aggregates (e.g., of aggregated distributed components) or of new components (e.g., of composite components). In this case, the world in which components live is typically formed by a set of localities, intended as the "solution" in which chemical reactions can occur, altough of course it is intended that components can flow/diffuse across localities to ensure globality of interactions.

Biological metaphors typically focusses on biological systems at the small scale, i.e., at the scale of individual organisms (e.g., cells and their interactions) or of colonies of simple organisms (e.g. ant colonies). The species are therefore either simple cells or very simple (unintelligent) animals, that act on the basis of very simple goal-oriented behaviours (e.g., move and eat) and that are influenced in their activities by the strength of chemical signals in their surroundings. Similarly to physical systems, in fact, components are expected (depending on their status) to be able to spread and diffuse (chemical) signals around, that can then influence the behaviour of other components. The laws of the ecosystem determines how such signals should diffuse, and how they could influence the behaviour and characteristics of components. The world in which components live is typically a virtual computational landscape that can influence the way signals diffuse and the way components can move over it.

Ecological metaphors focusses on biological systems at the level of animal species and of their interactions. The components of the ecosystem are sort of goal-oriented animals (i.e., agents) belonging to a specific species (i.e., agent classes), that are in search of "food" resources to survive and prosper (e.g., specific resources or other components). The laws of the ecosystem determine how the resulting "web of food" should be realised, that is, they determines how and in which conditions animals are allowed to search food, eat, and possibly produce and reproduce, thus influencing and ruling the overall dynamics of the ecosystem and the interaction among individuals of different species. Similarly to chemical systems, the shape of the world is typically organized around a set of localities, i.e., of ecological niches

	Species	***Laws***	***World***
Physical	Particles (computational components) and messages (computational fields)	Navigation and activities driven by fields (gradient ascent by components)	The Universe (a network), as shaped by waves and particles.
Chemical	Atoms (semantically described) and Molecules (composed semantic descriptions)	Chemical Reactions (matching of semantic descriptions and bonding of components)	Space (localities/bags of components)
Biological	Cells (amorphous computing cells, modules of self-assembly components)	Diffusion of chemical gradients and morphogens, differentiation of behaviour and activity	Space (Abstract computational landscapes, or physical landscapes)
Ecological	Organisms (Agents) and Species (Classes) and Resources (Data)	Survive (goal-orientation), eat, produce, and reproduce	Niches (Pervasive computing environments

Fig. 2. Metaphors for Service Ecosystems

(think at a set of local pervasive computing environments), yet enabling interactions and diffusion of species across niches.

3.2 Space, Time, and Control

The analysis of which metaphor to adopt cannot abstract from the fundamental question of: what do we want to achieve with it? What features do we want our ecosystem to express?

In general terms, as already stated in the introduction, we think that the ecosystem should be able to exhibit features of self-adaptation and eternity. Such features, from a very practical viewpoint, translate in: *(i)* the capability of the ecosystem of autonomously self-organize the distributed (i.e., spatial) activities of the components, so as to autonomously adapt the overall structure and behaviour of the system to specific contingencies; *(ii)* the capability of the ecosystem of tolerating changes over time, which includes the capabilities of adaptively accommodating new species or of surviving the extinction of species, as well as the capability of accommodating very diverse and composite behaviour with the same limited set of eco-laws. In addition, since we should never forget that the service ecosystem is here to serve us, we cannot forget an additional important feature, that is, *(iii)* the need allow humans (e.g., system administrator and users) to exert control over the behaviour of the ecosystem (or of some of its parts), i.e., of directing its activities and behaviour over space and time. All of these features, of course, should be enforced without paying the price of dramatically increasing the complexity of the ecosystem (i.e., the number and complexity of eco-laws, and the structure of its components and of the world in which they live).

The analysis of the extent to which the presented metaphors are able to accommodate (and how easily and naturally) the above features, is very complex, and would require much more room than the few pages of this paper. Nevertheless, we can try at least to draw some considerations about this, as summarised in Figure 3

Physical metaphors have been extensively studied for their spatial self-organization features, and in particular for their capability of facilitating the achievement of coherent behaviours even in large scale system (e.g., for load balancing and data distribution), and the conceptual tools available for controlling the spatial behaviour and the dynamics of such systems are well-developed. However, the physical metaphor seems to fall short in evolution and time adaptation, in that it hardly tolerates the presence of very diverse components with very diverse behaviours (at least if we want to preserve the simplicity of the eco-laws).

Chemical metaphors, on the other hand, can effectively lead to local self-organizing structures (e.g., local composite services) and, to a more limited extent, to some sorts of global structures (e.g., networks of distributed homogeneous components, as in crystals). Real chemistry, and so chemical computational metaphors, can accommodate an incredible amount of different components and composites, yet with the same set of simple basic laws. This is an important pre-condition for facilitating evolution over time. As far as control is concerned, one can think at using sort of catalyst or reagent components to control the dynamics and the behaviour of a chemical ecosystem.

Biological metaphors appears very flexible in enabling the spatial formation of localised morphological and activity patterns, and this has been shown to have notable applications in a variety of applications to distributed systems. However, the number of patterns that can be enforced by the spread of chemical gradients and by the reactions of simple individuals seem (as it is in physical metaphors) quite limited, and this does not match with the need for time evolution and adaption. Moreover, it is quite difficult to understand how to properly control the overall behaviour of such systems (just think at the fact that, so far, the mechanisms of morphogenesis are not fully understood by scientists).

Ecological metaphors, the same as chemical ones, promises to be very suitable for local forms of spatial self-organization (think at equilibria in ecological niches), and are particularly suited for modeling and tolerating evolution over time (think at how biodiversity has increased over the course of evolution, without ever mining the health existence of life in each and every place on earth). However, unlike chemical systems, understanding how to properly control the local and global equilibria of real ecological system is a difficult task, and it would probably be very difficult also in their computational counterparts.

In summary, it is very difficult to assess once and for all which of the metaphors is the best for next generation of adaptive service ecosystems. Some exhibit suitable features for certain aspects, but fall short for others. Personally, we have a preference for using the chemical abstraction as a basis—which seems to be the most flexible one—possibly extending it with features of other metaphors: hence the correct answer is probably in some new "hybrid" metaphor, getting the best of all the above.

	Space (self-organization)	***Time (evolution and adaptation)***	***Control (decentralized management)***
Physical	+ (global self-organizing spatial structures)	-- (no new components, always same behaviours)	++ (we know well how to build and control specific structures in physic)
Chemical	+ (mostly local self-organizing structures, sometimes global too, as in crystals)	++ (several new components can be generated under the same basic laws)	+ (reactants and catalysts can exert control over the dynamics and structure of reactions)
Biological	+ (local, morphogenesis of local shapes)	-- (limited number of new "shapes", and only local changes)	- (mechanisms of morphogenesis not fully understood)
Ecological	+ (local structures mostly, although sometimes leading to more global patterns)	++ (several new species and same laws)	-- (difficult to understand how to enforce control over ecologies of many species, at most only local centralized control)

Fig. 3. Advantages and Limitations of the Different Metaphors

4 Concluding Remarks

The peculiar characteristics of emerging and future network scenarios challenge current service frameworks, calling for novel service models and associated open service frameworks capable of exhibiting properties of autonomous adaptation and long-lasting (ideally eternal) availability and effectiveness.

In this paper, we have tried to elaborate on the idea of getting inspiration from natural ecosystem, i.e., of conceiving future service frameworks as an ecology of data, services and resources. There, services are modeled and deployed as autonomous individuals in an ecosystem of other services, data sources, and pervasive devices, and their interactions takes place in the form of a natural obedience to a simple set of well-defined "laws of nature". In this way, it is possible to deliver adaptivity and eternity as inherent properties of the service framework, rather than as complicated ad-hoc solutions.

Despite the promises of the ecological approach, though, the road towards the actual deployment of usable and effective "eco-inspired" open service frameworks still requires answering to several challenging questions. What metaphor, among the many possible ones (e.g., biological, physical, chemical) should be better adopted for modeling a suitable service framework? How should we model and represent individuals, the space in which they live, and the laws of nature to which they are subject? How can such individuals and the laws of nature lead to suitable, useful, and controllable forms of spatial self-organization? How can their dynamics be controlled to ensure eternal evolvability in an open setting? What shape should be taken by an actual software infrastructure that supports the ecosystem? All of these, and many further questions we may have missed identifying, open up fascinating areas of research.

References

1. Gul Agha. Computing in pervasive cyberspace. *Commun. ACM*, 51(1):68–70, 2008.
2. Ozalp Babaoglu, Geoffrey Canright, Andreas Deutsch, Gianni A. Di Caro, Frederick Ducatelle, Luca M. Gambardella, Niloy Ganguly, Márk Jelasity, Roberto Montemanni, Alberto Montresor, and Tore Urnes. Design patterns from biology for distributed computing. *ACM Trans. Auton. Adapt. Syst.*, 1(1):26–66, 2006.
3. Alistair P. Barros and Marlon Dumas. The rise of web service ecosystems. *IT Professional*, 8(5):31–37, 2006.
4. Jacob Beal and Jonathan Bachrach. Infrastructure for engineered emergence on sensor/actuator networks. *IEEE Intelligent Systems*, 21(2):10–19, 2006.
5. Gabriella Castelli, Alberto Rosi, Marco Mamei, and Franco Zambonelli. A simple model and infrastructure for context-aware browsing of the world. *Pervasive Computing and Communications, 2007.*, pages 229–238, 19-23 March 2007.
6. Jon Crowcroft. Toward a network architecture that does everything. *Commun. ACM*, 51(1):74–77, 2008.
7. Simon Dobson, Spyros Denazis, Antonio Fernández, Dominique Gaïti, Erol Gelenbe, Fabio Massacci, Paddy Nixon, Fabrice Saffre, Nikita Schmidt, and Franco Zambonelli. A survey of autonomic communications. *ACM Trans. Auton. Adapt. Syst.*, 1(2):223–259, 2006.
8. Sebastian Herold, Holger Klus, Dirk Niebuhr, and Andreas Rausch. Engineering of it ecosystems: design of ultra-large-scale software-intensive systems. In *ULSSIS '08: Proceedings of the 2nd international workshop on Ultra-large-scale software-intensive systems*, pages 49–52, New York, NY, USA, 2008. ACM.
9. Ramesh Jain. Eventweb: Developing a human-centered computing system. *Computer*, 41(2):42–50, 2008.
10. Mehdi Jazayeri. Species evolve, individuals age. In *IWPSE '05: Proceedings of the Eighth International Workshop on Principles of Software Evolution*, pages 3–12, Washington, DC, USA, 2005. IEEE Computer Society.
11. Jeffrey O. Kephart and David M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.
12. Marco Mamei and Franco Zambonelli. *Field-based Coordination for Pervasive Multiagent Systems*. Springer Verlag, 2006.
13. Maxim D. Peysakhov, Robert N. Lass, and William C. Regli. Stability and control of agent ecosystems. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1143–1144. ACM, 2005.
14. Raghu Ramakrishnan and Andrew Tomkins. Toward a peopleweb. *Computer*, 40(8):63–72, 2007.
15. Wei-Min Shen, Peter Will, Aram Galstyan, and Cheng-Ming Chuong. Hormone-inspired self-organization and distributed control of robotic swarms. *Autonomous Robots*, 17(1):93–105, 2004.
16. Mihaela Ulieru and Steve Grobbelaar. Engineering industrial ecosystems in a networked world. In *5th IEEE International Conference on Industrial Informatics*, pages 1–7. IEEE Press, 23-27 June 2007.