Mapping Ecology to Autonomic Communication Systems

Beatriz Otero[1], Pere Barlet-Ros[1], Salvatore Spadaro[2] and Josep Solé-Pareta[1]
Technical University of Catalonia, C/Jordi Girona 1-3,
Campus North, 08034, Barcelona Catalonia-Spain
[1] Computer Architecture Department
{botero, pbarlet , pareta}@ac.upc.edu
[2] Communications & Signal Theory Department
spadaro@tsc.upc.edu

ABSTRACT

IBM presented (in 2001) the idea of autonomic computing: many different ways of interacting. Self-governing components can simplify configuration, healing, optimization and protection of IT systems (thus hiding complexities to human operators). Today, Autonomic Technology also refers to the self-managing characteristics of resources as such the capacity of hiding completely its complexity to operators and users. Systems make decisions using high-level policies from operators. They will constantly check and optimize their status and automatically adapt themselves to changing conditions. Autonomic technologies may represent promising solutions for the evolution of Future Internet, ICT and Telecommunications.

This paper addresses the problem of designing future service frameworks based on autonomic technologies and leveraging bio-inspired laws, algorithms and patterns naturally existing in the ecology of living entities. We first define an overall architectural model and then the autonomic abstraction. Next, we study the principles and capabilities of autonomic communication systems based on ecological patterns and propose examples of how to map these laws into the architecture. This study is illustrated with two use cases that can be easily prototyped to test the feasibility of the model.

Categories and Subject Descriptors
C.2.1 [**Computer Systems Organization**]: Computer-Communications Networks – *Networks Architecture and Design;* K.4.1 [**Computing Milieux**]: Computers and Society – *General.*

General Terms
Design, Human Factors, Theory.

Keywords
Design principles, autonomic communication systems.

This work was partially funded by the Ministry of Science and Technology of Spain under contract TIN2007-60625 and by the HiPEAC European Network of Excellence and by the Catalan Research Council under contract 2005-SGR00481.

1. INTRODUCTION

Autonomic communication systems consist of interactive collections of individual autonomic components that contain resources and deliver services to humans and other autonomic components. These components manage their internal behaviour and their relationships with other autonomic components according to policies established by humans or other components of the system. Autonomic communication systems are complex information systems that comprise many interconnected components operating at different time scales in a largely independent fashion that manage themselves to satisfy high-level system requirements [16].

Ecology has always been a key source of inspiration for developing autonomic communication systems. Group-living animals have provided inspiration for the field of collective, or swarm intelligence, which models problems through the interactions of a collection of agents cooperating to achieve a common goal. In these systems, problems are "self-solved" in real time through the emergence of appropriate collective behaviour, which arises from the sum of all interactions occurring between the agents and their environment. The study of ecological processes and organisms can provide inspiration for extract models, principles or behaviour from ecosystems and apply them to autonomic communication systems [26]. Ecology offers significant knowledge that can be applied to different contexts [17] [19]. A small number of simple rules govern all ecological systems and produce patterns. These patterns can be used in autonomic communication systems for resource management, task allocation, etc.

In Figure 1, we propose a model that uses the ecology to define the interrelations between the users and "The World" (Web, devices, networks, etc.). Individuals (components) of different species interact, compete and combine with each other (with regard of basic laws that can be inspired from ecology) above a common environment substrate, so as to serve their own individual needs.

This paper is a preliminary attempt to understand the autonomic communication systems behaviour using one view point based on the ecology. On this hand, if we copy models and patterns from ecology, we will organised autonomic components and define the interactions between them. These will offer the possibility of create new algorithms frameworks.

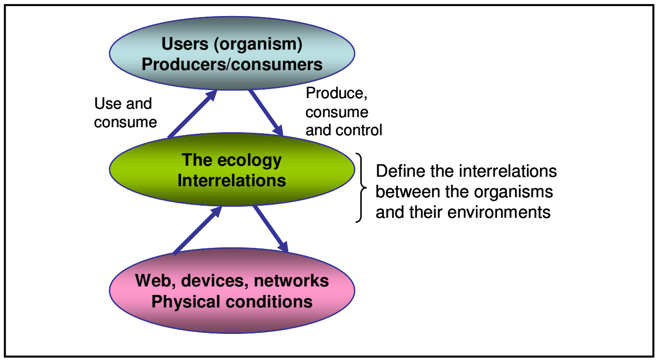

Figure 1. Autonomic communication system architecture

The remainder of the paper is organized as follows: Section 2 defines an autonomic component and describes their principles. For each principle, we cited one example. Section 3 describes the capabilities of autonomic communication systems and provides two examples for the self-organization capability. Section 4 shows two use-cases: Future Web and Real Time Cities. Finally, Section 5 has the conclusions of the paper and the future work.

2. ARCHITECTURE

2.1 Component Architecture

The similarity between ecology and a network of devices (grid) governed by autonomic principles is obvious: both are aggregations of components trying to maximize their performance, in an environment where efficiency is largely determined by the nature of their interactions with other components [20] [23]. These components form an ecosystem in which they help each other to select services which best match their preferences. Figure 2 shows how a community is formed by autonomic components, which are the smallest unit of an autonomic communication system. It is a self-contained system module with specified input/output interfaces and explicit context dependencies. We envision that the components will live in the ecosystem. Other examples of the community in the context of the autonomic communication system are the P2P communities.

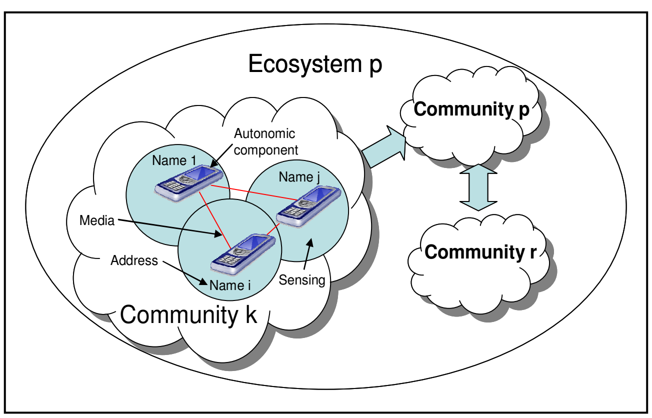

Figure 2. An example of ecosystem

In this paper, we adopt, as common modelling and treatment of services, data and devices, the model of the picture in Figure 3 which we extracted from [28]. For implementations based into multi-agents each module of this model will be an autonomic component's organ [9]. Then, in this case one autonomic component is composed of sets of interoperating modules called organs, where each of them has functionalities.

All "entities" living in the envisioned ecosystem will have an associated semantic representation. This is a very basic ingredient for enabling dynamic predictable interactions between components. Whatever autonomic component will be injected in the system and its definition or semantic representation will form part of the ecology.

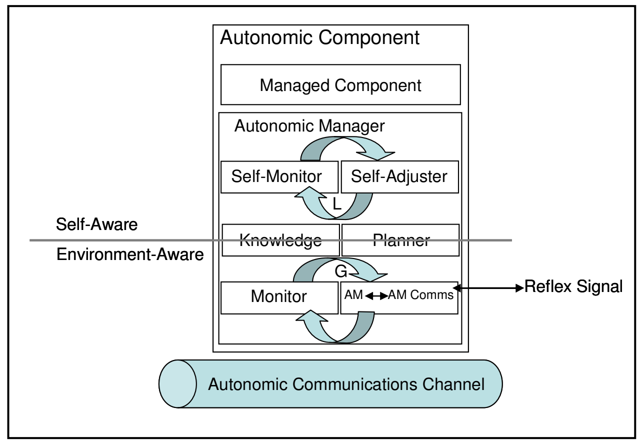

Figure 3. Abstraction of autonomic component [28].

2.2 Autonomic Principles

In general, the principles of autonomic communication systems are similar to the laws and patterns of ecology. If we apply the fundamental principles of ecology [25] to autonomic communication systems we obtain the following properties:

2.2.1 Compartmentalization

A compartment is defined by a set of members that are able and willing to communicate among each other according to compartment's operational and policy rules.

For example, compartments will allow decomposition of today's global IP network into appropriate sub-networks, which can be managed more autonomously from the overall network (e.g., a different addressing or routing scheme can be applied inside each compartment).

2.2.2 Atomization

The atomization is defined as the decomposition of systems into smaller and more easily manageable units. This principle also helps to reduce and hide complexity. Figure 4 shows an example of how the service execution can be developed in a highly distributed way. Executors are developed running over distributed resources (in peering) and are developed through autonomic components. An executor can be seen as a board that incorporates a number of workers, i.e. a board with execution capabilities. Equivalently, an executor acts toward other executors as a worker: it takes operations and transfers "tuples" to its board from overloaded executors [1].

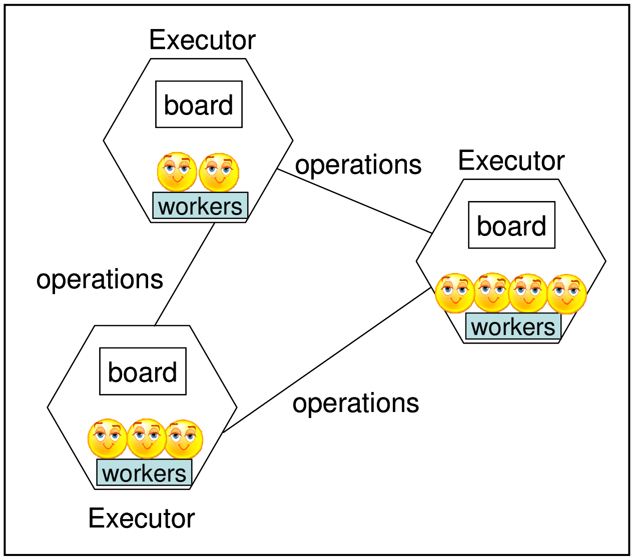

Figure 4. Atomic decomposition of execution environments

2.2.3 Re-composition
The principle of re-composition consists of a design that enables flexible, dynamic, and fully autonomic formation of large-scale networks in which the functionalities of each constituent network node are also composed in an autonomic fashion.

An example of this principle is the service composition where a suite of low-level services can be used to compose higher-level services.

2.2.4 Closed control loop
The closed control loop principle is related to the information that characterizes the dynamic behaviour of a system: growth, new life forms (biodiversity), mortality, etc.

For example, in autonomic networks this principle is used for measuring the parameters of the controlled system that are relevant for the function under control (e.g., current load for performance optimization, link availability for fault tolerance).

3. SELF-* ALGORITHMS
Autonomic communication systems must know their environment and the context surrounding their activity, and act accordingly [11]. They find and generate rules to best interact with neighbouring systems and adapt themselves. Autonomic interaction and policy-based self-management are technologies that support inter-component interactions, such as service-level agreements, negotiation protocols and algorithms, and conversation support.

3.1 Autonomic Capabilities
IBM, as part of its autonomic computing initiative, has outlined the need for current service providers to enforce adaptability and properties of self-configuration, self-optimization, self-protection and self-healing, via service (and server) architectures revolving around feedback loops and advanced adaptation/organization techniques. Sample self-managing in autonomic communication systems include: installing software when the system has detected that some software is missing (self-configuration), adjusting current workload when an increase in capacity is observed (self-optimization), taking resources offline when an intrusion is detected (self-protection) and restarting a failed component (self-healing) [20]. These properties are accompanied by four enabling properties or attributes, namely self-awareness, environment-awareness, self-adjustment and self-monitoring [10] [11] [16]. Since 2001 the self-* list of properties has grown substantially [29]. It now includes features such as self-anticipation, self-adaptation, self-definition, and self-organization among others [27] [29]. Driven by such vision, a variety of architectural frameworks based on "self-regulating" autonomic components has been recently proposed both by IBM and by independent research centres [4] [5] [8]. Moreover, [12] proposed a model to measure the degree of autonomicity (i.e. self-management) in these systems.

In recent years, a variety of diverse algorithms and approaches have proved the potential of eco-inspired distributed solutions to enforce purposeful functionalities in a fully distributed, self-organizing and adaptive way [2] [3] [9] [18]. Examples of these studies are: ant-inspired algorithms [7] [21] and socially-inspired communication mechanisms [13] [14].

3.2 Self-organization
Concerning approaches to model and build self-organizing frameworks, a variety of heterogeneous proposals exists for the design of the basic components and their interactions (e.g. pheromones [21] or gossip [15]). Proposals for the basic components include reactive agents rather than proactive and goal-oriented ones [30]. Other practical proposals study how to distribute and develop autonomic self-organizing services. For example, the Service Clouds [24] infrastructure combines dynamic software configuration methods with self-organizing algorithms for the establishment of communication links in order to support both cross-layer and cross-platform cooperation.

Other example of self-organizing framework is SwarmingNets [6]. This is a research framework for the management of complex Ubiquitous Service-Oriented Networks. In the SwarmingNets architecture, the required network service processes are implemented by groups of autonomic objects, called TeleService Solons as elements of TeleService Holons, analogue to individual insects as particles of the whole colony. A group of Solons have the capabilities of fulfilling the complex tasks relating to service discovery and service activation.

We provide two examples of self-organizing algorithms for the execution of services in a distributed environment: passive and on-demand clustering. In passive clustering, two nodes (components) are notified by a third one (match-maker) interconnected through an overlay network. Every node in the system has a chance of "waking-up" and assuming the role of "match-maker" to initiate a rewiring procedure. The match-maker randomly selects two of its own neighbours and, if they are of the same type, instructs them to link together. If the two chosen nodes were not already connected a new link is established between them.

On the other hand, on-demand clustering distinguishes between the initiator procedure of a rewiring procedure and the match-maker. More details of these algorithms can be found in [22].

4. USE CASE DEFINITION
Two demonstration use-cases have been already identified representing two aspects of the same evolutionary vision: Future

Web and Real Time Cities. The former one will demonstrate architectural features for overcoming limitations of the current HTTP client-server Web. A Future Web will be intrinsically exploited by the Open Self Eco Framework (OSEF) distributed architecture leveraging Semantic Web languages and standards (i.e., RDF, OWL, SWRL and SPARQL). The latter one will demonstrate architectural features of a Future Internet architecture capable of real-time monitoring the dynamics of a city through gathering and correlating data (anonymous localization, traffic, pollution, etc.), service components when they are produced, exposed and consumed. Such digital clouds can be used to provide situated services in order to better meet people's needs.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we studied the relation between the autonomic communication systems and the ecology. Both systems have a collection of autonomic components, which run intelligent control loops to analyze, monitor, plan and execute actions (functions) using knowledge of the environment. For this reason, we defined autonomic component and we described its laws and capabilities based on the ecology. In each case, we suggested examples and we illustrated how to apply autonomic principles in distributed environments. In particular, we presented two uses cases where the self-organization capabilities can be demonstrated.

As future work, we will design a prototype of ecology-based interrelations to experimentally evaluate a framework conceived as a goal-oriented world of components, interacting with each other for dynamic composition of data, knowledge and services in large-scale distributed environments. The framework design will need to study and evaluate a variety of decentralized self-* algorithms to enforce various forms of self-organization and self-composition of services by annotating semantically services and pieces of data and knowledge. This prototype will offer the possibility to identify new algorithmic solutions for the dynamic and semantic goal-oriented composition of data and services.

6. ACKNOWLEDGMENTS

Authors would like to thank A. Manzalini (Telecom Italia) for the fruitful discussion and collaboration on the matter.

7. REFERENCES

[1] Alfano R. and Di Caprio G. (2006) "TURBO: An Autonomous Execution Environment with Scalability and Load Balancing Features". Workshop on Distributed Intelligent Systems: Collective Intelligence and its Applications (15/06/2006). Vol. 00, pp. 377-382, ISBN 0-7695-2589-X.

[2] Baboglu O., et al. (2007). "Design Patterns from Biology to Distributed Computing". ACM Transaction on Autonomous and Adaptive Systems, Vol. 1, No. 1, pp. 26-66. DOI: http://doi.acm.org/10.1145/1152934.1152937.

[3] Bernadas A., Manzalini A., and Moiso C. (2008). "Demonstrating Communication Services Based on Autonomic Self-organization". Proceedings of the Second International Conference on Complex, Intelligent and Software Intensive Systems (CISIS).

[4] Cai G., Gao J. and Hu B. (2006). "A Lightweight Approach towards Autonomic Service-Oriented Computing System Development". Proceedings of the IEEE Asia-Pacific Conference on Services Computing (APSCC'06), pp. 392-399.

[5] Chen Y. et al. (2007). "SLA Decomposition: Translating Service Level Objectives to System Level Thresholds". Proceedings of the Fourth International Conference on Autonomic Computing (ICAC'07), pp. 3-3. DOI: http://doi.acm.org/10.1109/ICAC.2007.36.

[6] Chiang F. and Braun R. (2006). "A Nature Inspired Multi-Agent Framework for Autonomic Service Management in Pervasive Computing Environments". Proceedings of the 10th IEEE/IFIP Network Operations and Management Symposium, NOMS'06, pp. 1-4.

[7] Chiang F., Braun R. and Agbinya J. I. (2007). "Self-Configuration of Networks Services with Biologically Inspired Learning and Adaptation". Journal of Network and Systems Management, Vol. 15, No. 1, pp. 87-116. DOI: 10.1007/s10922-006-9056-3.

[8] Farha R., Kim M., Leon-Garcia A. and Won-Ki Hong J. (2005). "Towards an Autonomic Service and Architecture". Lecture Notes in Computer Science 3751: 58-67. DOI: 10.1007/11567486.

[9] Fusco A. et al. (2008). "Autonomic Wireless Communications in Digital Cities: An Experimental Use Case". Proceedings of the Mobilware. Austria.

[10] Ganek A. G. Corbi T. A (2003). The Dawning of the Autonomic Computing Era. IBM System Journal 42(1):5-18.

[11] Horn, P. 2001. Autonomic Computing: IBM Perspective on the State of Information Technology. IBM T. J. Watson Labs., NY. DOI: http://www.research.ibm.com/autonomic/. (Accessed May 2008).

[12] IBM. An Architectural Blueprint for Autonomic Computing (3rd. ed.), White Paper, 2005.

[13] Jelasity M., and Babaoglu O. (2005). "T-Man: Gossip-based Overlay Topology Management". Proceedings of Engineering Self-Organising Applications (ESOA'05).

[14] Jelasity M., Guerraoui R., Kermarrec A. M., and Van Steen M. (2004). "The Peer Sampling Service: Experimental Evaluation of Unstructured Gossip-based Implementations". Proceedings of the Fifth ACM/IFIP/USENIX International Conference on Middleware. LNCS 3231, pp. 79-98. DOI: 10.1007/b10156.

[15] Jelasity M., Montresor A., and Babaoglu O. (2005). "Gossip-based Aggregations in Large Dynamic Networks". ACM Transactions on Computer Systems, 23(3):219-252.

[16] Kephart J. O. and Chess D. M. The Vision of Autonomic Computing. Computer (2003), 36(1):41-52.

[17] Lodding, K. N. 2004. The Hitchhiker's Guide to Biomorphic Software. ACM Queue 2, 4, 66-75.

[18] Mamei M., Menezes R.,Tolksdorf R. and Zambonelli F. (2006). "Case Studies for Self-organization in Computer Science". Journal of Systems Architecture, Vol. 52, pp. 8-9.

[19] Ottino, J. M. 2004. Engineering Complex System. Nature No. 427, 399. DOI: 10.1038/427399a.

[20] Parashar M. and Hariri S. (2005). “Autonomic Computing: An Overview”. International Workshop on Unconventional Programming Paradigms, Mont Saint Michel, FRANCE (15/09/2004). Vol. 3566, pp. 257-269, ISBN 3-540-27884-2.

[21] Parunak H. V. D. (1997). “Go to the Ant: Engineering Principles from Natural Multi-Agent Systems”. Annals of Operations Research, 75: 69-101.

[22] Project CASCADAS: Work Package 3. Aggregation Algorithms, Overlay Dynamics and Implications for Self-Organized Distributed Systems. http://www.cascadas-project.org/index.html. (Accessed May 2008).

[23] Saffre F, Halloy J. and Deneubourg J. L. 2005. “The Ecology of the Grid”. Proceedings of the Second International Conference on Autonomic Computing (13-16 June, 2005). Seattle, WA, USA. IEEE Computer Society, 378-379. ISBN 0-7695-2276-9.

[24] Samini F. A., et al. (2007). “Service Clouds: Distributed Infrastructure for Adaptive Communication Services”. IEEE Transactions on Network and System Management, Special Issue on Self-Managed Networks, Systems and Services.

[25] Scheiner S. M and Willing M. R. A General Theory of Ecology. Theory Ecological (2008) 1:21-28. http://www.springerlink.com/content/3652561862q77592/fulltext.pdf

[26] Stanford Encyclopedia of Philosophy. DOI: http://plato.stanford.edu/entries/ecology/. (Accessed April 2008).

[27] Sterritt R. (2005). Autonomic Computing. Innovation System Software 1(1). DOI: 101007/s11334-101005-0001-5.

[28] Sterritt R., Parashar M., Tianfield H. and Unland R. (2005). A Concise Introduction to Autonomic Computing. Advanced Engineering Informatics 19, pp. 181-187.

[29] Tianfield H (2003). Multi-Agent Based Autonomic Architecture for Network Management. Proceedings IEEE International Conference on Industrial Informatics, INDIN, (21-24 August), pp. 462-469.

[30] Tummolini L., Castelfranchi C., Ricci A., Virola M., and Omicini A. (2005). “Exhibitionists and Voyeurs do it better: A Shared Environment Approach for Flexible Coordination with Tacit Messages”. Environments for Multi-Agent System, LNAI Vol. 3375.