A Systematic Review of Self-adaptation in Service-oriented Architectures

M. Pilar Romay
Dept. Inf. & Com. Syst. Engineering (DISIT)
St. Paul-CEU University
Boadilla del Monte, Madrid, Spain
pilar.romayrodriguez@ceu.es

Luis Fernández-Sanz, Daniel Rodríguez
Dept. of Computer Science
University of Alcalá (UAH)
Alcalá de Henares, Madrid, Spain
luis.fernandezs@uah.es, daniel.rodriguezg@uah.es

***Abstract*—The study of adaptivity, i.e., the capability to react to changes in the environment, is becoming ever more important in many fields of study, and in the development of software in particular. This paper presents a systematic review in which both the extension and complexity of this notion are examined. After studying the influence from external fields, this review checks the hypothesis of using the scope of service-oriented architecture as a comparable model for the whole field. As part of the systematic review, the influence of the most relevant bibliography is considered, and the terminology is clarified.**

Keywords – adaptivity; self-adaptation; service architecture; autonomic systems; SOA; systematic review.

I. INTRODUCTION

The growing complexity, along with continuous operation, of software systems –not only conventional ones, but also the next complexity level, so-called *large-scale software systems* [1] – has greatly increased the interest of a series of techniques for self-managing system features. These techniques make possible for them to guarantee a wide range of properties, all by themselves. Traditionally, these properties had been dealt with manually, or had to be developed from specific requirements. Instead of that, the new approach considers them as *intrinsic* system properties, and thus they should be dealt with automatically, and considered as just another issue in conventional software systems development.

Systems conceived in such a way are generically known as *adaptive systems* or, more specifically, as *self-adaptive systems* [2].

Therefore, we have systems able to deal with faults and critical situations (*self-healing*), able to control their own behavior (*self-managing*) [3], able to observe and evaluate their own performance (*self-monitoring*), able to modify their own configuration to react to changes in their environment (*self-configuring*), or even to automatically guarantee certain system-level properties, such as protection, fault tolerance, etc. (*autonomic systems*) [4] [5], among many others.

The wide range of systems which could make use of these *adaptive properties* causes a great variability; therefore many different approaches could be conceived. For this reason, it is reasonable to focus our efforts on a specific area of study: in our case, *software services*. This area has been chosen because it still covers a wide range of systems and shows a great variability itself, and therefore it can be considered as a representative, even a lower-scale analog, for the whole of the field of adaptive systems. The goal of this work is, therefore, to study *self-adaptive software services*.

Software services define, due to their own properties, an area of a great potential to describe and use adaptive (or adaptation-related) features. Moreover, *service* and *service-oriented architectures* are among the systems where the need for these features is clearer, and more compelling: the nature of services is inherently dynamic, and this implies the need for adaptation; and also the structure of service architectures requires the flexibility that self-adaptation provides. In short, this make our specific goal (consider adaptation in services, rather than in general systems) even more pragmatic. Finally, considering the growing, relevance and broad dissemination of *service ecosystems*, this is also the environment in which this approach is currently pertinent and more interesting.

Adaptivity is often described at different levels, namely at *service* level or the wider *system* level [6], but this will not be the main interest of our study. Instead of that, we will focus on exploring and analyzing adaptivity and all its related properties, a set which is often generically known as *self-**.

Moreover, this paper will also consider the impact of *self-organization* (considered as a related notion, rather than as an adaptive feature) within the specific area of service-oriented architecture. Our main goal is to determine which properties are implied in adaptive systems, with a special focus on service architectures – i.e., to be able to *evaluate adaptivity* in services.

For this purpose, this paper presents an initial study of the field, which will be used to delimit the boundaries of the area and to check the reliability of the hypothesis about the service-oriented approach and its applicability to evaluation. The core of this study is structured as a systematic review: after defining a set of goals and the corresponding research questions, and discussing the background on the field, the review makes an extensive bibliography review, which is carefully examined and analyzed in order to achieve the corresponding conclusions.

The paper is structured as follows: first, we present the context of our study, including the definition of four primary goals and the method of our systematic review. Then, we provide some background justifying the interest of this study, as well as the implicit connections between its areas. After that, we characterize the revised information, and outline the method we have used to locate and classify this information, describing the performed searches and their results. We end by summarizing the conclusions from several perspectives.

II. CONTEXT OF THE STUDY

Though it might seem a secondary issue, the relevance of adaptivity is such that even well-known authors as Kramer & Magee have claimed [7] that “a significant advance in the techniques which are required for the effective development of adaptive systems would imply an advance of an order of magnitude in every fundamental aspect of Software Engineering”.

Having this relevance in mind, the main goal which has driven the conception of this study is focused in finding a *model* which makes possible, by using a set of attributes, to define and assess *adaptivity* in the context of *services*. This model could alternatively take the form of a framework, or even a methodology.

Then, this paper intends to provide a characterization of the field of adaptivity. For this purpose, it lays out a set of specific *goals* to drive the study, which should make possible to measure and digest the breadth of the field, and to confirm the need of the study itself. Moreover, it also intends to value the most important contributions in the process.

From these goals, the paper follows a methodological approach based on [8] with the purpose to achieve a greater soundness than a traditional narrative description. The more important limitations of such a study are also considered: *publication limitations* (publishing bias), and *selection limitations* (selection bias). The first one refers to the relative impact of *negative* studies –i.e. which have not significant differences with previous proposals–, when these have not been published, or are only rarely referenced in the literature. Also, there could be interesting studies written in another language, or even duplicate references which could later influence the *metanalysis*. The second one is related to the definition of inclusion and exclusion criteria: the purpose is to avoid to neglect the inclusion of relevant work, and also to include misleading papers, which hinder dealing with the relevant topics in an objective way.

The systematic review will be developed in the next sections. First, we will describe the main goals of the study, and outline our methodological approach. Next, we will briefly explore the background on service-oriented architectures and their relationship to adaptivity, focusing on the need for evaluation and the influence of dynamic service composition models. Then the systematic review itself is unfolded: after presenting the main data sources, the search strategies and selection criteria are described – to later present the results of the review and discuss the conclusions.

A. Goals of the Study

Therefore, the goals of our systematic study are: (1) To confirm the breadth and applicability range of adaptivity. (2) To verify the novelty of this field of study within the context of Software Engineering. (3a) Related to the previous one, to evaluate adaptivity in service-oriented architectures. (3b) To assess interesting contributions which could be applied to the study’s primary goal (i.e., to determine the properties which characterize adaptive systems). (4) By exploring the previous four points, to identify the used terminology.

B. Methodological Approach in the Study

This (systematic) study begins by planning the review, then conducting the review, and finally reporting the review. The first activity of this process is a bibliographic search. Based on a set of *research questions* related to context definition, modelling, and management, we defined a list of *keywords* and search strings used for our investigation.

The defined searches will be oriented to cover the goals of the study, as proposed in section II.A. In this part of the process, the selected keywords and their synonyms are of a great relevance: the obtained results strongly depend on a good selection of these terms.

For this reason, we also designed and realized an specific search, focusing on articles and papers which tried to provide a wider vision of the field, such as (other) research reviews, overviews, state-of-the-art articles, etc.

The initial terminology search should just be considered as an approximation, and it will be later tuned and adjusted, to be refined by means of the obtained results during all the process. To some extent, the process itself serves as the main *control* in this initial search phase, and it could cause an additional iteration within the systematic review process – it just depends on the actual extension and variability of the terminology in the field.

After the search, we proceed to select and evaluate the obtained information. For this purpose, as already noted, the study defines acceptance and rejection criteria related to its specific goals, and in particular to the main goal – which was the reason to do the study, in the first place.

To finish, the obtained results will be analyzed and interpreted. In this phase, the process will make possible to synthesize the results with regard to the proposed goals.

III. BACKGROUND: ADAPTIVITY IN SERVICE-ORIENTED ARCHITECTURES

Nowadays, the notions of service orientation (or *service-oriented computing*, SOC) [9] and service-oriented architectures (SOA) have been totally integrated in the current conception of software. This is the reason why they define a perfect workbench to assess adaptive properties in generic software systems.

Therefore, this section reviews the context of work in both fields, focusing in the assessment of adaptivity, and service-oriented architectures.

A. Adaptivity Assessment & Evaluation

Adaptive systems can be defined as “systems able to react to automatically adapt themselves to changes in their environment”. This reaction can be specifically programmed, or could rise from an *emergent* behavior. This category of systems has been globally designated with the name of *self-* systems* [10], which explicitly refers to the variability of the concrete aspect to consider. However, in recent times most authors prefer to designate them with the generic name of *adaptive* (or *self-adaptive*) *systems*, like this paper did also in the Introduction.

Adaptivity is a complex field of study. First, because the term has explicitly been conceived to be generic, so we must first decide which specific feature ("attribute") are we going to consider every time. And second, because too often we lack a clear reference model which could serve as the basis to compare to the system's degree of adaptivity. Therefore, it is necessary to have some kind of *model* to make possible to assess those capabilities, either quantitatively (in the ideal case) or at least roughly, by approximation.

In general, the development of adaptive systems, as well as the more concrete development of *adaptive services*, lacks a clear set of methods and metrics able to assess the actual capabilities of a specific implementation. That is, it is really difficult to even decide if a given system is "adaptive" or not. In fact, this particular distinction is almost intuitive; however the increasing importance of this features, and the difficulties in their implementation, highlight the relevance of achieving the definition of a quantitative approach, able to deal with concrete values. For this reason, one of the main goals of this study focuses in checking existing references, which enable or guide the process to obtain either the model or relevant metrics, to be able to assess the level of adaptivity, even in a qualitative way.

B. Service-Oriented Architecture: Service Composition Mechanisms

A well-known definition of service-oriented architecture (SOA), as given by Michael Papazoglou [11] states that it is "a *meta-architectural style*, based in loosely coupled services, which provides flexibility to business processes in an interoperable way, and independently from the technology". Therefore, its main goal is *interoperability*, which is itself a consequence of loose coupling.

However, a standard definition of SOA is still debated, in spite of the popularity of the term – probably because it has been used with different meanings in different contexts, and referring to different technological aspects. Beyond those details which distinguish the many variants of the concept of *service* (web services, RESTful services, grid systems, etc.), there are still several intrinsic features in its definition. These features imply that service-oriented architectures are *a priori* more dynamic and flexible than many "traditional" ones, in particular component-based architectures – and this can be considered inherent to its own nature.

From this point of view, it is interesting to note at least two of these features, which suggest this kind of architecture as a good evaluation workbench for adaptivity:

1) External Composition Mechanisms. First, services are always part of a modular system – they are conceived to be used as part of a larger structure. However, there is a subtle difference to more traditional approaches: service systems are designed to be composed at runtime.

Services cannot assume anything about the rest of the elements in the composition. First, their interface is separated from the rest of the service, and therefore services never interact *directly* to the rest of the system. Second, they are not designed as part of a concrete compound: instead of that, once they are implemented and deployed, they are included in some composite system, which was *later* conceived.

These are the reasons why the well-known composition models for services (choreography and orchestration) must be conceived as external compositions. Thus a service does not even need to know if it is contained in a composite: the business logic (the "intelligence" of the system) belongs in the structure itself, not in its individual components. Within an *orchestration*, it is in the orchestrator; but *choreographies* are even more complex, as the composition schema is distributed along the composite – i.e. it is *decentralized*. Every individual service receives just a "local" subset of instructions, without a perspective of the global plan. Even *service mashups*, a promising approach, are again an external composition model – in fact, essentially an orchestration.
Another consequence is *crosscutting*. Unlike traditional composition, service models do not preclude that *the same* service is simultaneously a part of more than *one* composite. This implies that every service composition is *orthogonal* to any other which is performed later [6].

2) Instrinsically Open Architecture. Of course, many existing systems, and distributed systems in particular, have claimed to define an open architecture. In practice, an *open system* is every system which, by defining or using an standard interface, is able to compose any external element defined as a client of that interface. However, if constraints imposed by this interface are too strict, the limits they define hinder the capture of information about the different clients – i.e. it would present an homogeneous architecture,which is exactly the opposite of our goal.

Services use a different approach: the interface is defined at the beginning, to offer a concrete functionality (a service), and to guarantee a certain quality level (i.e. QoS). But apart from that, services are conceived, even at the technical level, to be composed to *any* other element able to interact to them. Therefore, they are presented as the ultimate *open* system: in the specific case of RESTful web services, for instance, the only actual constraint is the use of the HTTP protocol, which was conceived using the REST architectural style itself - and this is not an actual constraint, nowadays.

Also, we have to consider that the current evolution of service systems has a clear trend towards a significant rise of the *scale*. The original "XML web services" were in general small modules, of a scale comparable to that of objects, or even smaller. Currently, the concept is clearly shifting to be equivalent to so-called *Software as a Service* (SaaS) – where the scale of a service is similar to that of a complete application. In fact, the approach itself is evolving from the potential provided by a concrete technology which focused on interoperability, to the design of a new, generic software distribution model (shifting from "product" to "service").

In any case, current *service-oriented architectures*, when this term is understood in the wider sense [12], present the same features of flexible and open composition we have already noted – and this makes them adequate as a workbench for *adaptivity evaluation* in software systems.

IV. Revised Information & Methodology

This study is based on information obtained from several digital bibliography search engines. Specifically, we have used search engines from the best known and most widely recognized publishers in the fields of Computer Science & Information Technology, as well as Google Scholar.

To have a preliminary structuring of the area, we first considered the results provided by Google Scholar. The goal was to assess the research activity on *adaptive systems* in the period 2000-2011, including *every* potential environment, and comparing these results to those in the specific subarea of service-oriented architectures in section A .

The remaining searches followed a more systematic approach, guided by specific goals (in the form of questions), specifically those which were proposed in section II.A.

Throughout all this search process, we have considered the possibility of evaluating the used terminology, with the purpose of extending the search to a wider scope – but still within the parameters of the study. This evaluation has made possible to change and evolve the initial searches, to the final form we will describe in the following.

After performing those search processes, our inclusion and exclusion criteria were used to select the most relevant articles. Then we also examined the references cited in these papers, with the purpose to select other relevant papers, which were not located previously due to their publication stage, or which have been published by some additional publisher. This way, the publishing bias we mentioned in section II.B.

A. Search Strategy and Selection of Areas

The search strategy has been guided by our goals, by answering to a set of questions.

The questions were bound to specific terminology. The variety of meanings of some of the terms used in our search made necessary to apply an iterative, evolutionary approach, in which those search terms were finely tuned. At the end of the process, our study has made possible to obtain a specific terminology summary, which covers goal (4). This specific terminology, obtained from multiple sources in the revised information, has been represented using a pyramidal mesh, which will be detailed in section IV.C. Therefore the most significant terms and notions related to our field of study have been collected, also emphasizing their similarities and differences, something which is not always completely clear. This way, in our iterative process we have refined concepts such as *autonomic* vs. *autonomous*, *adaptation* vs. *self-adaptation*, *adaptive*, *self-organization*, *self-monitoring*, etc.

The definition of these terms, as part of the results for our goal (4), is briefly explained in section IV.C, where it also explains the aforementioned pyramidal structure.

Within these terms, we should emphasize those which were considered for our search, namely:

- Adaptation, adaptive, adaptivity, self-*
- "Software service", service-oriented, SOA
- Evaluation, "quality model"

In order to fulfill our first and second goals, we performed a series of searches on Google Scholar, as well as other databases. In the final search on Scholar, the questions related to these goals were the following:

- Assessment of the number of articles dealing with adaptivity, against the number of those doing the same in the service-oriented architecture area.
- Which disciplines (research areas) are dealing with and applying adaptation?

The first search, which intends to identify the different fields of study related to adaptivity, is driven by the following queries, referring to the compared subsets:

- Query #1: (("autonomic" OR adaptive OR adaptation OR autonomous OR adaptivity OR self) AND (evaluation OR quality))
- Query #2: (("autonomic computing" OR adaptive OR adaptation OR autonomous OR adaptivity OR self) AND (evaluation OR quality)) AND (("software service") OR ("service-oriented Architecture") OR ("Service Oriented Architecture"))

These queries, on the Google Scholar engine, resulted in about 7.806.600 references for query #1 and a total of 17.565 for query #2. The refined search provides roughly about 1500 results every year, from 2000 to 2011. The scope of the study is very wide, covering almost any scientific area – which is not surprising and confirms our intuition.

2011	2010	2009	2008	2007	2006	2005	2004	2003	2002	2001	2000
770	2980	3180	3050	2630	1980	1410	668	379	243	189	86

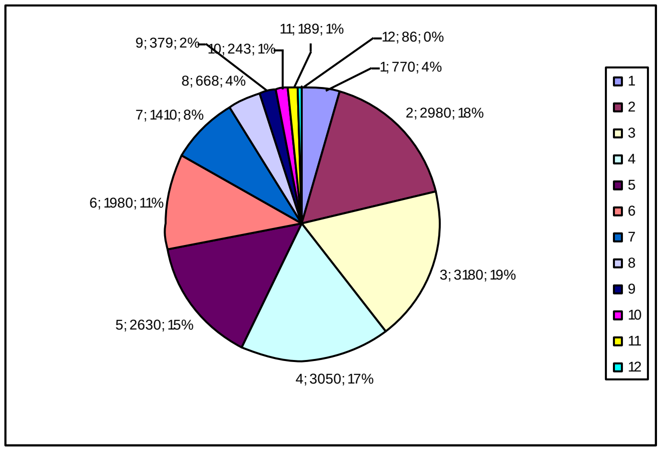

Figure 1. References for Query #2 from 2000 to 2011 (1-12)

The most representative areas for query #1 are: Life Science, Engineering, Social Science and Law, Mathematics and Statistics, Medicine and Computer Science (e.g. Ubiquitous Computing, Grid Environments [13], mobile systems and services [14] [15], Domotics [16], etc.)

Goals numbered as (3) are essential in the context of this study – i.e. the evaluation of adaptivity in service-oriented architectures. Related searches have been more specific, and they have already been performed in bibliography databases from the publishers themselves. The purpose was to obtain a more accurate list of articles, trying to reach all the relevant information – without any accidental loss. We also have used references from the selected articles, and in relevant cases we have also searched for the corresponding citations.

For instance, a representative query could be:

- Query #3: ((("autonomic computing" or self) and (adaptive or adaptation or adaptivity)) and (evaluation or quality)) and (("software service") or ("service-oriented Architecture") or ("Service Oriented Architecture"))

B. *Inclusion and Exclusion Criteria.*

1) Related to goals (1) and (2): Neither inclusion nor exclusion criteria were defined – this search was delimited just by query clauses themselves, i.e. queries #1 and #2. This could seem less "systematic" than the remainder of the study. However, we did not intend to do a detailed classification of areas and fields of study, but to assess if our suggestion (to focus on service architectures) was reasonable. This goal alone could be used to justify a specific systematic study, which would be even more complex than the one presented here. The reason to include this goal is to perform a shallow examination of some of the areas suggested by many search engines, with the purpose of perceiving the actual extension of the field, as well as its growing rate. A systematic study on this specific aspect would be of great interest to detect methods or tools (from other fields) which could be applied in the context of adaptive software.

2) Related to goals numbered as (3). In this search process, queries are quite more specific, and they mainly focus in evaluating adaptivity by means of self-properties. Therefore it considers papers including models, frameworks, metrics and evaluations on the topic. This study excluded papers not dealing with self-properites, and those which did not focus on assessing adaptivity/autonomic features.

3) Related to goal (4): The resulting terminology has been extracted from papers selected in the previous phase. Therefore their inclusion and exclusion criteria are the same. However, some additional selection criteria are also added; specifically, articles which define or clarify terminological aspects, or which perform reviews in which terminological features are also clarified.

C. *Results*

1) Related to goals (1) and (2): The range and scope of the many fields of study which apply adaptivity is too wide to be considered in this paper – in fact, it would require an specific study itself. Therefore, for this purpose we refer to the results outlined in section IV.A, and to the conclusions summarized in sections V.B and V.C, which expose a global vision for this part of our study.

2) Related to goal (3): This goal, together with results about terminology from goal (4), provides a characterization of adaptivity. The following table summarizes briefly this part of the study. It describes representative categories of existing work, indicating for each one of them references, goals, projects, metrics and their organization.

TABLE I. EVALUATION OF ADAPTIVITY

Ref	**Evaluation of Adaptivity**	
	Goal/ Project/ Context	***Metrics/ Organization***
[28] [29]	Metrics to evaluate Self-* systems criteria / -- / Web-based C/S, E-learning (AHA!), Videoconference, Multiagent Systems	The many metrics for each Propierties (reuse, genericity…) /methodological, architectural, intrinsic characteristic and runtime
[30]	Metrics for restarting strategies in WS Reliable Messaging (WSRM) / -- / WSRM	Effective Transmission Time (ETTi), Unnecessary Resource Consumption (URCi), Savings (SAVi) / Adaptation parameters (structures, payoff, environments, time)
[34]	Quality Model for the software architecture of self-healing applications (based on ISO 9126) /Attribute-based architectural styles (ABAS) / --	Traditional quality attributes (Maintainability –Modifiability, Extensibility-, Reliability –Fault tolerance, Robustness-) Specific Autonomic Quality attributes (Support for detecting anomalous system behavior, Failure Diagnosis, Simulation of expected behavior, Differencing between expected and actual behavior, Testing of correct behavior). Autonomic Metrics: Detection ratio, Detection time, Fault Model Observability, Awareness, Coupling / Traditional and Autonomic attributes
[35]	User-level Quality of Service (QoS) (Context awareness) / PLASTIC, model PFM / Pervasive Networking Environment	Performance evaluation
[36]	Quality model to evaluate Self-* attributes (adopts 6 features of ISO 9126: Reliability, Efficiency, Maintainability, Usability, Functionality, Portability) / -- / --	The autonomic maturity of each level in complex software (Complexity of development, business domain and management) / Three-level Autonomic Evaluation Model (Software Complexity, Relative Quality Factor, Autonomic features). Fuzzy comprehensive evaluation (qualitative factors)

3) Related to goal (4): These results are summarized in Fig. 2, which shows the wide spectrum of so-called self-properties, ranging from very generic properties which can be applied in many systems (such as context-awareness) to specific attributes which are only found in some approaches (like emergence). Apart from these, there are several other, less frequent, properties – also, many of them are referred to using different names and variants (self-managing vs. self-management). All these issues have been considered in the study, and they are implicitly included in this paper.

Fig. 2 represents *three* pyramids rather than one – they are conceptually related, but they must be studied separately. Pyramid #1 represents *environmental adaptation*, i.e. the capability of a system to perceive its own environment and integrate in it. Pyramid #2 represents *behavioral adaptation*, i.e. the capability of a system to modify its behavior to adapt to different conditions, ranging from pure observation to full self-management. And pyramid #3 depicts *self-adaptation*, i.e. the capability of the system to manage its own adaptivity, possibly including its own emergent behavior. Together, this triple representation describes the full range of *adaptation*.

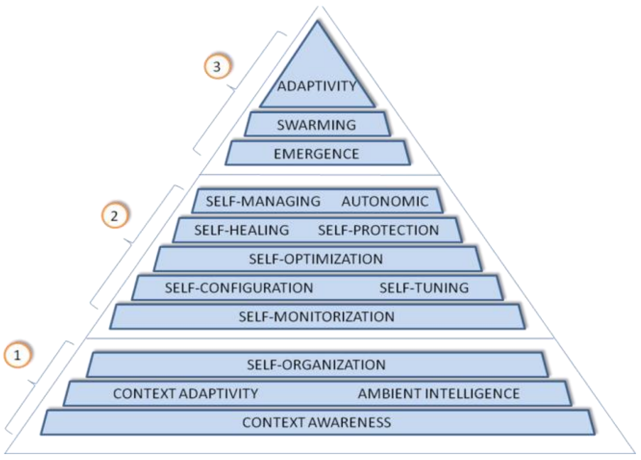

Figure 2. The spectrum of self-properties: a pyramidal representation

This (triple) pyramid represents a *gradient*, rather than strict layers – i.e. each level is more complex than the one below itself (at least inside its own pyramid), but it is not necessarily using its services, though it is probably supported by some of the layers below. The same applies to the three pyramids – their separation depicts a gradient, but they can be considered independently. For example, an autonomic system is in the cusp of pyramid #2 – this means it is more complex than a self-healing system, but not necessarily that there is an emergent behavior (from pyramid #3) above it.

Therefore the pyramidal representation must not be understood literally – its purpose is to give an idea of their relative conceptual scope and size. As noted, some of these properties are built on top of the previous step (for instance, self-management should always rely on self-healing), but this is not always true (for instance, self-organization is not necessarily based on context adaptation).

This representation also helps to outline the distinction between similar but different terms: for instance, *adaptation* (i.e. the full range in the triple pyramid) vs. *self-adaptation* (i.e. just the range in pyramid #3). A similar conflict appears to differentiate *autonomous* (i.e. the capability of a system to act independently) from *autonomic* (understood here as the combination of several self-properties [4][22]). Indeed, there is an intimate relationship between adaptation and autonomy; though they describe different features, to fully achieve each one of them, the other is also required, at least partially.

V. CONCLUSIONS

We present our conclusions in the following, structured as the next four sections.

A. *Adaptivity and self-properties*

Regarding adaptation, there are significant differences in the way in which these autonomous changes in the system must be performed. This is mainly related to the way they are managed [17]. The range covers from the *ad hoc* way, in which adaptation (or the adaptors) needs the intercession of some stakeholder [18], to the *automatic* way, in which adaptation (and the adaptors) is fully generated by tools [19].

Self-organization can also be studied within the context of adaptivity [20], as we have already done in the previous section (Fig. 2). It should be considered nevertheless as an independent property, with the same level or complexity and interest than adaptation itself – of course, the same applies to the evaluation process [21]. This feature can also be considered in relation to several self-properties (such as self-adaptation or self-assembly, in particular), though it is more basic (and at the same time, can be more complex) than the majority of the properties listed in Figure 2. This reflection also requires a discussion of the terminology.

In many cases, the evaluation of adaptivity needs to have into account the specific context to deal with – some systems require to be adaptive even when their flexibility is minimal. This relative scale must also be considered.

B. *Adaptivity in different areas*

The wide scope of the field suggests that there could be methods and techniques designed for the evaluation of adaptivity [2] [4] [5] [22] which could be applied at the software architecture level. Several techniques have also been inspired in other fields, such as the Control Loop Model [2], and some others can still be transferred – much of them in the context of natural systems, in particular in the context of self-organization.

The growing relevance of this field is even more apparent in the context of "new" kinds of applications which are appearing right now and in the near future. An obvious example is adaptation in the context of *mobile systems*, where context-awareness, which includes a wide range of techniques, has been an active line of research.

C. *Adaptivity in Software Engineering*

An immediate conclusion, with respect to the field of software engineering, is that the evaluation and assessment of adaptivity is still a relatively new area. A review of the existing literature shows that there are still several aspects to define, such as languages or methods [23][24][25], etc. Once this is done, the quality of service (QoS) could be influenced by adaptivity, just like it is now by interoperability – this would be used as the criteria to select and use certain systems [17]; in summary, this could provide soundness to autonomous systems. There is already some amount of work in this direction, but these are still proposals under discussion, the first contributions which must be refined.

Adaptive systems also begin to be considered within the specific subfield of Requirements Engineering, for instance [26]. But, besides deciding when to adapt (adaptation time), we are also interested in the nature of adaptive capabilities, and how to define generic models which could determine our adaptive systems.

D. Evaluation of Adaptivity

In summary, we can conclude that currently there is not any effective method able to evaluate the adaptivity of a software system [27][28] [29][30][31][32][33][34][35][36] – not even when we refer to this property not in the wider sense, but focusing on a concrete feature.

Also, as deduced from section IV.B, the scope of service-oriented architecture is comparatively much smaller than the general scope of adaptivity. But while the size of the field has maintained constant, the importance of services has increased – therefore, we can conclude that our hypothesis is reasonable, and then, that adaptive services can be used as a model for generic adaptivity.

REFERENCES

[1] L. Northrop, Ultra Large-Scale Systems: The Software Challenge of the Future, SEI Books, Software Engineering Institute, 2006.

[2] B.H.C. Cheng, R. de Lemos, H. Giese, P. Inverardi, and J. Magee, Software Engineering for Self-Adaptive Systems, Lecture Notes in Computer Science 5525, Springer, 2009.

[3] B.H.C. Cheng, *et al*, "Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2010)," 32nd International Conf. on Software Engineering (ICSE 2010), ACM Press, 2010, pp. 447-448.

[4] J.O. Kephart and D.M. Chess, "The Vision of Autonomic Computing," IEEE Computer, vol. 36, 2003, pp. 41-50.

[5] M.C. Huebscher and J.A. McCann, "A Survey of Autonomic Computing -- Degrees, Models and Applications," ACM Computing Surveys, vol. 40, 2008.

[6] C.E. Cuesta and M.P. Romay, "Elements of Self-Adaptive Systems - A Decentralized Architectural Perspective," Self-Organizing Architectures, Lecture Notes in Computer Science, vol. 6090: Springer, 2010, pp. 1-20.

[7] J. Kramer and J. Magee, "Self-Managed Systems: an Architectural Challenge," Future of Software Engineering (FoSE 2007), IEEE Computer Society, 2007, pp. 259-268.

[8] B. Kitchenham, "Procedures for performing systematic reviews," Joint Technical Report, Keele University and National ICT, 2004.

[9] M.P. Papazoglou and D. Georgakopoulos, "Introduction to the Special Issue on Service-Oriented Computing," Communications of the ACM, vol. 46, 2003, pp. 24-28.

[10] O. Babaoglu, M. Jelasity, *et al*, Self-star Properties in Complex Information Systems: Conceptual and Practical Foundations, Lecture Notes in Computer Science 3460, Springer, 2005, pp. 1-20.

[11] M.P. Papazoglou, Web Services: Principles and Technology, Prentice-Hall, 2007.

[12] NEXOF Reference Architecture Specification, Version 1.0, 2010. http://www.nexof-ra.eu/?q=node/695. Last access: 07/15/2011.

[13] D. Ardagna, S. Lucchini, R. Mirandola, and B. Pernici, "Web Services Composition in Autonomic Grid Environments," Business Process Management Workshop, Springer, 2006, pp. 375-386.

[14] R. Rouvoy, P. Barone, Y. Ding, F. Eliassen, S.O. Hallsteinsen, J. Lorenzo, A. Mamelli, and U. Scholz, "MUSIC: Middleware support for self-adaptation in ubiquitous and service-oriented environments," Software Engineering for Self-Adaptive Systems, Springer, Lecture Notes in Computer Science 5525, 2009, pp. 164-182.

[15] G. Wrzesinska, J. Maassen, and H.E. Bal, "Self-adaptive applications on the grid," Proceedings of the 12th ACM SIGPLAN symposium on Principles and practice of parallel programming, New York, NY, USA: ACM, 2007, pp. 121-129.

[16] J. Ferreira, J. Leitão, and L. Rodrigues, "A-OSGi: A Framework to Support the Construction of Autonomic OSGi-Based Applications," Autonomic Computation and Communication Systems: Autonomics 2009, Lecture Notes of the ICST 23, Springer, 2010, pp. 1-16.

[17] C. Canal, J.M. Murillo, and P. Poizat, "Software Adaptation," L'Objet: logiciel, bases de données, réseaux, vol. 12, 2006, pp. 9-31.

[18] C. Peper and D. Schneider, "On runtime service quality models in adaptive ad-hoc systems," 2009 ESEC/FSE workshop on Software Integration and Evolution @ runtime, ACM, 2009, pp. 11-18.

[19] A. Bottaro and R. Hall, "Dynamic Contextual Service Ranking," Software Composition, Springer 2007, pp. 129-143.

[20] M. Randles, A. Taleb-Bendiab, and D. Lamb, "Cross Layer Dynamics in Self-Organising Service Oriented Architectures," Self-Organizing Systems, Springer, 2008, pp. 293-298.

[21] L. Liu, S. Thanheiser, and H. Schmeck, "A Reference Architecture for Self-organizing Service-Oriented Computing," Architecture of Computing Systems (ARCS 2008), U. Brinkschulte, T. Ungerer, C. Hochberger, and R. Spallek, eds., Springer, 2008, pp. 205-219.

[22] P. Lin, A. MacArthur, and J. Leaney, "Defining Autonomic Computing: a Software Engineering Perspective," Proceedings Australian Conference on Software Engineering (ASWEC 2005), IEEE Computer Society Press, 2005, pp. 88-97.

[23] M. Wolski, C. Mazurek, P. Spychała, and A. Sumowski, "The architecture of distributed systems driven by autonomic patterns," Software Engineering Techniques: Design for Quality, K. Sacha, ed., Springer Boston, 2007, pp. 49-60.

[24] Y. Liu, M. Tan, I. Gorton, and A. Clayphan, "An Autonomic Middleware Solution for Coordinating Multiple QoS Controls," Service-Oriented Computing (ICSOC 2008), A. Bouguettaya, I. Krueger, and T. Margaria, eds., Springer, 2008, pp. 225-240.

[25] D. Menasce, H. Gomaa, S. Malek, and J. Sousa, "SASSY: A Framework for Self-Architecting Service-Oriented Systems," IEEE Software, Early Access article, IEEE, in press.

[26] K. Welsh and P. Sawyer, "When to Adapt? Identification of Problem Domains for Adaptive Systems," Proceedings of the 14th international conference on Requirements Engineering: Foundation for Software Quality, Springer-Verlag, 2008, pp. 198-203.

[27] J.A. McCann and M.C. Huebscher "Evaluation issues in autonomic computing". Proceedings of Grid and Cooperative Computing Workshops (GCC), IEEE CS Press, 2004, pp. 597-608.

[28] L. Masciadri and C. Raibulet, "Frameworks for the Development of Adaptive Systems: Evaluation of Their Adaptability Feature Through Software Metrics," 4th International Conference on Software Engineering Advances (ICSEA 2009), 2009, pp. 309-312.

[29] C. Raibulet and L. Masciadri, "Evaluation of Dynamic Adaptivity through Metrics: an Achievable Target?," Joint Working IEEE/IFIP Conference and European Conference on Software Architecture (WICSA/ECSA 2009), IEEE CS Press, 2009, pp. 341-344.

[30] P. Reinecke, K. Wolter, and A. van Moorsel, "Evaluating the adaptivity of computing systems," Performance Evaluation, vol. 67, 2010, pp. 676-693.

[31] D. Robinson and G. Kotonya, "A Self-Managing Brokerage Model for Quality Assurance in Service-Oriented Systems," High-Assurance Systems Engineering, IEEE, 2008, pp. 424-433.

[32] G. Feuerlicht, "Simple Metric for Assessing Quality of Service Design," Service-Oriented Computing, Springer, 2011, pp. 133-143.

[33] Luqi and G. Jacoby, "Testing Adaptive Probabilistic Software Components in Cyber Systems," Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems (16th Monterey Workshop), LNCS 6662, Springer, 2011, pp. 228-238.

[34] S. Neti and H.A. Muller, "Quality Criteria and an Analysis Framework for Self-Healing Systems," Proc. Software Engineering for Adaptive and Self-Managing Systems (SEAMS'07, ICSE), IEEE Computer Society, IEEE Digital Library, 2007, p. 6.

[35] M. Autili, P. Inverardi, and M. Tivoli, "Run Time Models in Adaptive Service Infrastructure," Run-time mOdels for Self-managing Systems and Applications, Springer, 2010, pp. 125-152.

[36] H. Zhang, H. Whang, and R. Zheng, "An Autonomic Evaluation Model of Complex Software," International Conference on Internet Computing in Science and Engineering, 2008, pp. 343-348.