On Adaptivity: Developing a Metric for an Elusive Concept

Philipp Reinecke
Humboldt-Universität zu Berlin
Institut für Informatik
Berlin, Germany
preineck@informatik.hu-berlin.de

Katinka Wolter
Humboldt-Universität zu Berlin
Institut für Informatik
Berlin, Germany
wolter@informatik.hu-berlin.de

Abstract

The ever-growing complexity of today's computing systems renders manual reconfiguration infeasible. Instead, systems must be able to adapt themselves to changes in their environment. Thus adaptivity becomes a key feature of a system. However, so far no general metric for the comparison of systems according to their adaptivity has been established. In this paper we identify the two dimensions of adaptivity, review a number of approaches and then seek to define a sufficiently generic metric.

I. Introduction

Today's computing systems grow ever more complex, and hence manual (re)configuration becomes less feasible. This results in an increasing interest in adaptive systems, systems that autonomously modify their behaviour to perform satisfactorily within their environments. For these systems, adaptivity must be considered among the key features of a system.

However, despite several interesting approaches in various fields, no general metric to assess a system's adaptivity exists. In fact, even the concept of 'adaptivity' itself is often subject to varying interpretations. In particular, the concept is sometimes used to describe the act of external adaptation, i.e. manual reconfiguration of a system, which is inconsistent with the goal of making systems more autonomous.

Consequently, our development of an adaptivity metric must start with a definition of the concept. The next step is the formalisation of the problem, followed by a review of existing approaches in Sections II and IV. We will then propose our metric, concluding with a brief discussion of its characteristics and further work.

As our general approach tends to be rather abstract, we will employ a restart oracle within a Web Services Reliable Messaging (WSRM) implementation as an illustrative example throughout the text. Put briefly, WSRM provides delivery guarantees for the communication between Web Services [2], [6], [7]. A restart oracle within the implementation computes retransmission timeouts, i.e. intervals between restarts of the task of transmitting a message.

A. Adaptivity

We define adaptivity as the *ability of a system to adapt itself to its environment*. Adaptation occurs with the aim of delivering optimal performance. We can identify two dimensions of adaptivity:

1) Starting from its initial state, the system must be able to adapt itself in such a way that it provides an optimal level of performance. That is, it must change its internal state in order to optimally exploit the features of the environment that confronts it. In this view, a static environment is assumed.
2) In any state, the system must be able to react to changes in the environment.

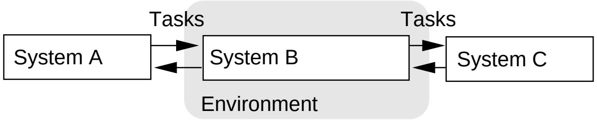

Fig. 1. Generic system-environment view.

II. THE PEtS SPACE

In a general way, we can view a system within its environment as depicted in Figure 1: The system B acts as a server to another system A, i.e. it receives and fulfills certain tasks. In completing a task it delegates some part of the task to other system(s) C. As seen from A, task completion has a number of properties that determine how 'useful' it is to A, the most prominent being completion time $T_C \in [0, \infty)$ (where $T_C = \infty$ amounts to a failure of B).

Both system A and system(s) C form the *environment* within which B operates and which influences B's task completion properties. An adaptive system modifies its behaviour, i.e. its interactions with A and C, to optimise its properties, as perceived by A. The system's changes are determined by knowledge obtained about the environment. Typically, B's knowledge is limited to what it can infer from observing the streams of tasks to A and C. In particular, B can often not observe the properties of task completion that are important to A directly, and usually it cannot observe internal behaviour of C.

Drawing from the treatment in [5], we refer to the part of the internal state of B that determines its behaviour within the environment as its *structure*, with system behaviour over time, over environments or over both forming a sequence of trials of structures. A payoff function encodes how useful the structure tried at time t in the environment E is to A, i.e. P maps observations of the metrics that describe task completion properties to $\mathbb{R}$. The current position in E (i.e. the current environment) can change at any time, out of the systems' control, and usually without explicit notice. Adaptation refers to the choices of structure S the system makes; optimal adaptation means that the system always selects the structure with the highest payoff within the current environment.

We combine these dimensions in the PEtS space (Figure 2). For simplicity, we will assume that all dimensions can be described by scalars, e.g. environments E can be enumerated and drawn along a single line.

Given a system B, we can observe its behaviour within this space, that is, we have a function

$$Obs : \{t\} \mapsto P \times E \times S$$
$$Obs(t) := (P(t), E(t), S(t))$$

that describes the trajectory taken by B throughout our observation interval $\{t\}$. That is, every value for $Obs(t)$ encodes the environment $P(t)$, the structure $S(t)$ tried within this environment and the payoff $P(t)$ obtained by this trial, all at time t. Note that this observation function implies an omniscient observer not subject to the limitations of B.

A. Example: A WSRM Oracle

If we consider a WSRM restart oracle as the system B, system A is the upper-layer application, whereas C consists of the SOAP Transport and the network stack beneath. The task fulfilled by B is the reliable delivery of a message using the (unreliable) system C. To ensure message transmission, B may need to restart the transmission task delegated to C, i.e. resend the message. Non-functional properties of task completion are the single-trip Effective Transmission Time (ETT), i.e. the time required for a reliable transmission, and the fairness, which can be measured by the Unnecessary Resource Consumption (cf. [6]). A restart oracle has only limited knowledge about these properties and about the network stack's behaviour. In particular, it can infer only round-trip times (RTTs)

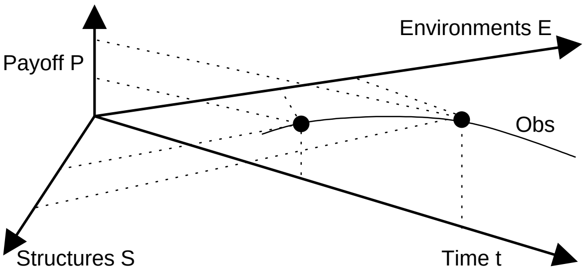

Fig. 2. The PEtS space.

from observations of message sending times and acknowledgement arrival times; and it can deduce unfair behaviour by counting the number of restarts.

Within the PEtS space, the *structure* of a WSRM oracle is its choice of a restart timeout τ. *Environments* can be described by e.g. the percentage of packet loss at the IP layer or by a characterisation of the workload generated by the application. *Payoff* is a weighted sum of ETT and URC, for instance:

$$P = \frac{1}{1 + \alpha ETT + (1-\alpha)URC} \qquad \text{(for } \alpha \in [0,1]),$$

which has the benefit of being bounded to $[0,1]$. The maximum at 1 reflects that the (albeit theoretical) optimum for (ETT, URC) lies at $(0,0)$.

III. Desirable Properties of a Metric

Metrics for adaptivity are defined using metrics for the performance of system B, as measured from system A's point of view. An adaptivity metric should possess a number of properties:

1) Boundedness: The values of the metric should be bounded, e.g., for a system S:

$$Ad(S) \in [0,1].$$

This property ensures that even single values carry some amount of information about the system, without the need for comparison to other values. For instance, the availability:

$$A = \frac{MTBF}{MTBF + MTTR}$$

has this characteristic.

2) Comparability: Given two values of the metric for two systems, there should be a simple way of comparing them. Scalar values (e.g. availability), in particular, are easily compared to each other. However, it should be noted that compression into a scalar value inevitably loses information.
3) Intuitive Interpretation: The metric should be obtained in such a way that its values can be interpreted easily by someone without a strong background in a specific modeling technique. Again, availability provides a good example for this property.
4) Simple and Efficient Computation: On-line computability of the metric is also desirable, since then the adaptivity of a system under specific circumstances can be monitored. This offers the option of exchanging systems on the fly, should their adaptivity prove insufficient.

IV. Approaches in the PEtS Space

Basically, any assessment of adaptivity amounts to some form of analysis of the observation function Obs, which, in order to ensure boundedness and intuitive interpretation, is often set in relation to the behaviour of an optimally adaptive system (either theoretical or practical). In the following we will discuss several approaches, starting with concepts limited to certain dimensions of the PEtS space and then addressing two more general fields.

A. Robustness

In [1], Ali et al. propose to use the *robustness radius* r^*, that is, the smallest amount of deviation from the nominal operation point in any parameter that causes the system to leave its robustness region (the region in E where system performance P is within acceptable limits) as a measure for the system's robustness.

r^* is computed using the impact function ϕ, which describes the impact perturbation parameters have on the performance features of the system. In our terminology, this function can be written as $\phi : E \mapsto P$, from which we see that ϕ only takes into account changes in the environment. ϕ must be obtained by observing an actual system's reactions to changes in the environment, which amounts to an averaging of the structures and time dimensions.

B. Optimal Allocation of Trials

Concentrating on the structure dimension instead, the Optimal Allocation of Trials (OAT), as discussed in [5], refers to adaptation in a static environment. Here, the structures dimension is treated as a multi-armed bandit: Each trial of a structure results in a certain payoff, and the system strives to accumulate the highest payoff possible. However, since it does not know the distribution of payoffs among structures, it has to operate within the tradeoff between exploitation and exploration. I.e., at each trial B has to decide whether to try the structure with the highest observed payoff (thus exploiting knowledge obtained so far) or to try another structure, which might yield higher payoff (exploration).

According to [5], given an ordering of structures reflecting their average payoff the optimal number n^* out of N trials to assign to the observed best structure can be estimated. Then, the ratio of actual trials allocated to the observed best structure to n^* is a metric for the adaptivity.

In practice, this requires sampling Obs to obtain the ordering of the structures, which implies averaging the environment dimension (the time dimension can be safely ignored if we assume that P is time-invariant).

C. Payoff Accumulation

Still assuming a static environment, but now focusing on the time dimension, the payoff accumulation rate [5] is another metric for the adaptivity. Given the payoff P^* an optimal system would have obtained at any time t, adaptivity is measured by the ratio between actual accumulated payoff and optimal accumulated payoff in the observation interval $[0, T]$:

$$Ad_1 = \frac{\int_0^T P_B(t)\mathrm{dt}}{\int_0^T P^*(t)\mathrm{dt}}.$$

An obvious extension of this metric would be to include changes in the environment, yielding

$$Ad_2 = \frac{\int_{\{t\}} \int_E P_B(e,t)\mathrm{dedt}}{\int_{\{t\}} \int_E P^*(e,t)\mathrm{dedt}}.$$

D. *Control Analysis and Sensitivity Analysis*

In contrast to the first three approaches, both control analysis and sensitivity analysis take into account all dimensions of the PEtS space, that is, they study system behaviour over changes in the environment, the structures and time.

In sensitivity analysis, the focus lies on identifying parameters that influence system output, and on finding points in the parameter space around which the system either behaves optimally or critically. Control analysis is largely concerned with identifying the system's response to certain inputs; where the response is considered in terms of stability, control accuracy, settling time and overshoot [3], [4].

Both fields share one common trait in that they analyse a model of the system in question. That is, based on Obs, system behaviour is approximated by a set of functions, which are then analysed, e.g. control analysis may model system behaviour by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k), \end{aligned}$$

where $x(k+1)$ describes the next structure, $y(k)$ describes the payoff and the matrices A, B, C encode system behaviour. To obtain these matrices, one again has to observe Obs and fit a model to the observations.

E. *Discussion*

Reviewing these approaches in regard to our initial two dimensions of adaptivity, we note that the OAT metric and the first payoff accumulation metric measure adaptation in the first sense, whereas the robustness radius only considers changes in the environment. Control analysis and sensitivity analysis study both adaptation to a static environment and within changing environments. The second payoff accumulation metric also considers changing environments, however, being an integral, it only provides an average value.

Considering the desirable properties for a metric, we note that most of these approaches do not exhibit all of them: The robustness radius offers an intuitive interpretation and comparability, but is neither bounded nor easily computed, since it relies on knowledge of the impact function ϕ. The OAT metric requires an ordering of structures, which can only be obtained through prolonged observation of Obs; furthermore, the procedure to estimate n^* works only for large total numbers of trials. Lastly, both sensitivity analysis and control analysis do not lend themselves to intuitive interpretations easily, since they rely on extensive modeling. Thus results for models with the same model structure (but different parameters) are comparable, while differing model structures (e.g. higher-order difference equations in control analysis) might introduce factors that hamper comparability of the results.

V. Metric Proposal

We attempt to work within the dimensions P, E, $\{t\}$ and S directly. We study the quality of the system's decisions at each trial $i = 1, 2, \ldots, N$ and use the probability of beneficial decisions as a measure of adaptivity. In the following, we assume that the payoff function is bounded to $[0, 1]$, with 1 representing the optimal payoff.

We consider the trajectory described by the observation function Obs and order the observations by t, so that i refers to the ith trial, occuring at time t_i. This trial of a structure $s_i = S(t_i)$ yields a payoff $p_i = P(t_i)$. Whether p_i is lower, equal to or larger than the payoff for the previous trial, $p_{i-1} = P(t_{i-1})$, shows how good the system's decision at this trial was: $p_{i-1} > p_i$ reflects a bad decision, since the system chose a structure that yielded a worse payoff than the one in the previous trial. $p_{i-1} = p_i$ amounts to a neutral decision. Finally, $p_{i-1} < p_i$ indicates a positive decision, i.e. the system chose a structure that increased payoff. Let

$$\begin{aligned} I_{\odot} &:= \{i | p_{i-1} = p_i\} \\ I_{\oplus} &:= \{i | p_{i-1} < p_i\} \end{aligned}$$

denote the set of neutral and positive decisions, respectively.

Clearly, positive decisions are beneficial. Neutral decisions are beneficial as well, since they refer to trials where the system was able to maintain its level of payoff. However, the benefit obtained in a trial must be quantified. The benefit of a neutral decision is equal to p_i, e.g. a constant payoff of 0 is certainly not beneficial at all, while $p_i = 1 \forall i$ would be optimal. A positive decision's benefit is equal to the payoff increase $\Delta_i = p_i - p_{i-1}$, i.e. large increases are more beneficial than small ones.

We propose to use

$$Ad_3 := \frac{\sum\limits_{i \in I_{\oplus}} \Delta_i + \sum\limits_{i \in I_{\odot}} p_i}{N - 1}$$

as a metric for the adaptivity of the system.

VI. Further Work

We argue that the metric Ad_3 has all the desirable properties and also expresses both dimensions of adaptivity in one scalar value. However, some aspects may need further investigation:

Payoff changes can be caused by a new choice of structure, a new environment or both. So far the metric does not fully take into account these possibilities. In particular, decreased payoff due to deteriorating environmental parameters is considered negative. Possible implications for its applicability need to be discussed in more detail.

The metric only considers one-step changes. It has to be considered whether this limitation has undesirable consequences. Namely, adaptation may require accepting small short-term losses to obtain large long-term gains, however, in the current definition acceptance of short-term loss is punished.

Finally, the usefulness of the metric must be shown through its application in a practical setting. While the algorithm for the metric as defined above is straight-forward, the two previous paragraphs point at possibly necessary modifications that may render it more difficult.

References

[1] S. Ali, A. A. Maciejewski, H. J. Siegel, and J.-K. Kim. Measuring the robustness of a resource allocation. *IEEE Transactions on Parallel and Distributed Systems*, 15(7):630–641, 2004.

[2] BEA Systems, IBM, Microsoft Corporation Inc, and TIBCO Software Inc. Web Services Reliable Messaging Protocol (WS-ReliableMessaging), February 2005.

[3] D. G. Cacuci. *Sensitivity & Uncertainty Analysis, Volume I: Theory*. CRC Press, 2003.

[4] J. L. Hellerstein, Y. Diao, S. Parekh, and D. M. Tilbury. *Feedback Control of Computing Systems*. Wiley Interscience, 2004.

[5] J. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1975.

[6] P. Reinecke, A. P. A. van Moorsel, and K. Wolter. The fast and the fair: A fault-injection-driven comparison of restart oracles for reliable web services. In *QEST '06: Proceedings of the 3rd International Conference on the Quantitative Evaluation of Systems*, pages 375–384, Washington, DC, USA, 2006. IEEE Computer Society.

[7] The Apache Software Foundation. Apache Sandesha. http://ws.apache.org/sandesha/.