# Towards Self-Regulating Adaptive Systems

**Alexandros Paramythis**
Institute for Information Processing and Microprocessor Technology
Johannes Kepler University, Linz
Altenbergerstr. 69, A-4040 Linz, Austria
alpar@fim.uni-linz.ac.at

## Abstract

This paper discusses ongoing work towards a theoretical basis intended to facilitate the development of self-regulating adaptive systems. Self-regulation refers to the capacity of the system to assess the effects of, and modify, its own adaptive behaviour in prescribed ways at run-time. Although not new, the concept of self-regulation is largely missing from existing adaptive systems, arguably due to the perceived complexity involved in its theoretical grounding and practical implementation. The paper addresses in particular the following two questions: What are the operational requirements of self-regulating adaptive systems? What implications does self-regulation impose on the modelling- and decision making- approaches used? The theoretical benefits of "clusters" of self-regulating systems, and the role of human experts in the self-regulation process are also briefly discussed.

## 1 Introduction

The concept of "self-regulating" adaptive systems was proposed in the late eighties by [Trevellyan and Browne, 1986] and was introduced in a taxonomy of adaptive systems by [Totterdell and Rautenbach, 1990]. The rest of this section will provide an informal definition of self-regulation in adaptive systems, and discuss its potential benefits, as well as the factors that have had detrimental effects in its employment in adaptive systems.

To start with, along the lines set out in [Trevellyan and Browne, 1986] and [Totterdell and Rautenbach, 1990], it is argued that effecting self-regulation in adaptive systems requires that the later be capable of *learning*. This specifically entails that the adaptive system be capable of (incrementally) modifying the "knowledge"[1] it uses for deciding upon adaptation. Such learning would basically result in adapting the system's own adaptive behaviour, to better accommodate different users, situations, environments, etc. This, in turn, would necessitate the capability, on the part of the adaptive system, to assess its own adaptive behaviour and determine whether it has met its goals (or, in other words, whether it has had the desired effects) and act accordingly.

To understand the motivation behind self-regulation in adaptive systems, let us first consider the typical operation of "traditional" adaptive systems. Firstly, adaptive systems create a model of their environment, which involves at the very least the system's user, and may also incorporate other dynamic and static information affecting interaction (e.g., the context of use). Secondly, the system's adaptation "logic"[1] (embodied in rules, Bayesian decision networks, neural networks, etc.) correlates the model(s) of the system's environment with a range of adaptive behaviours that the system is capable of. An important point to note is that what we referred to as the system's adaptation "logic" is never updated dynamically / automatically (or, at least, not without human intervention). An obvious benefit of this approach is that the system's adaptive behaviour is predictable. This also constitutes, however, the weakest point of traditional adaptive systems: adaptation logic is never "questioned", and is applied "blindly" (i.e., irrespectively of whether it actually achieves the desired effects or not).

The main goal behind self-regulation, then, is to enable adaptive systems to progressively validate, and where necessary, modify their own adaptive behaviour in prescribed ways. Totterdell and Rautenbach [1990] also argue that the levels of adaptivity reflect a change of intention moving from a designer specifying and testing the mechanisms in a (simple) adaptive system, to the system itself dealing with the design and evaluation of its mechanisms in a self-modifying system. The most obvious, and perhaps simplest, modification a self-regulating system can apply to itself is to "demote" (the use of) adaptation logic that does not have the desired effects.

Since self-regulation bears such great promise, why is it then that it has not yet proliferated in adaptive systems? The answer is two-fold: On the one hand, self-regulation *is* part of some adaptive systems in wide use today in different guises (e.g., recommender systems which use implicit and explicit user feedback to modify their recommendation strategies), albeit in rather restricted forms. On the other hand, as Benyon [1993] points out, moving up the levels of adaptivity incurs an increasing cost, which may not be justified. The most prominent cost in employing complete approaches to self-regulation is the inherent requirement for self-evaluation. Furthermore, there do not exist, to date, proposals on how self-regulation can be formalised and applied across the wide range of approaches to modelling and decision-making, common in adaptive systems today.

[1] The term "knowledge" is used here to generically refer to the whatever combination of modelling- and decision making- approaches are employed by the adaptive system to achieve adaptive behaviour, and does not imply that a "knowledge-based" approach is used to that extent. Similarly, the term "logic" is used to refer generically to the decision making approach, rather than the employment of logic-based reasoning, etc.

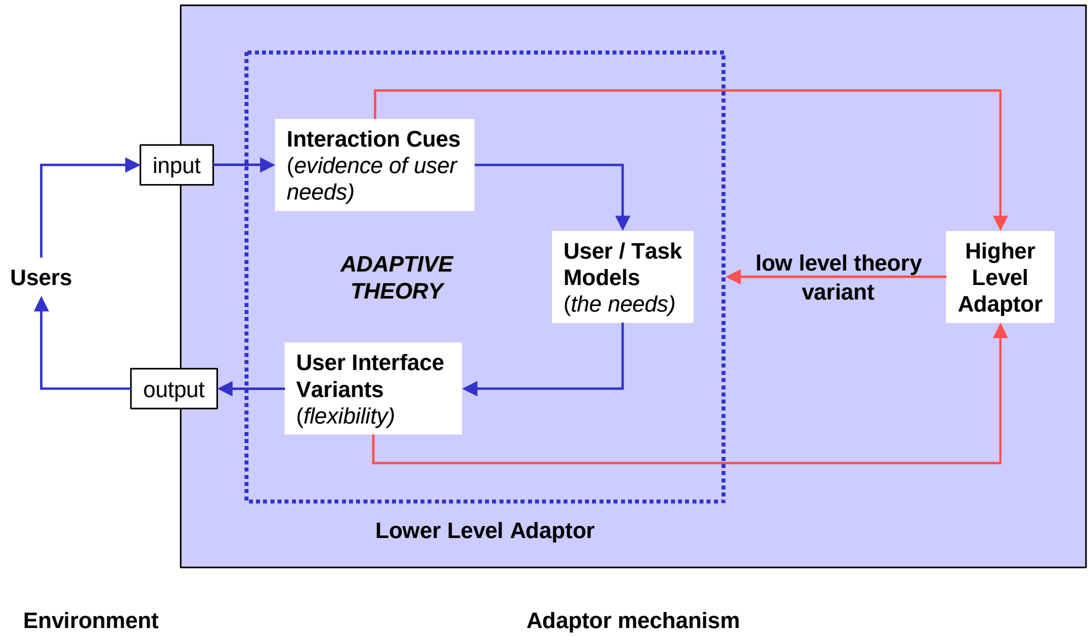


**Figure 1:** Logical diagram for a two-level adaptation architecture; adapted from (Totterdel and Rautenbach, 1990).

The rest of this paper discusses the main premises of a theoretical basis intended to facilitate the development of self-regulating adaptive systems. Due to lack of space the discussion is informal, does not go into detail, and necessarily leaves out topics that some might consider pivotal to self-regulation. The topics that are discussed include the operational requirements of self-regulating adaptive systems, and the implications of self-regulation in relation to modelling and decision-making. The paper is concluded with a brief discussion of the theoretical benefits of "clustering" self-regulating systems, and the role of human experts in the self-regulation process.

## 2 Dissecting Self-Regulation

### 2.1 Operational Requirements

In general, self-regulation in adaptive systems requires what has been termed a "two-level adaptation architecture" (Totterdel and Rautenbach, 1990), depicted schematically in Figure 1. This type of architecture requires that the system have two adaptation foci: one for adapting its interactive behaviour, and one for adapting its adaptive behaviour. The input to the "first-level adaptor" comprises user interactions and may include information provided explicitly by the user, characteristics of the context of use (as conveyed by appropriate "sensors"), etc. The output of the "first-level adaptor" consists of modifications effected to the system, which directly or indirectly affect user interaction with the system. The input to the "second-level adaptor" comprises that of the "first-level adaptor" and, in addition, the actual modifications applied to the system as a result of first-level adaptation logic. Although not depicted in Figure 1, this input may also encompass (or, actually, be entirely composed of) information from the various system models. The output of the "second-level adaptor" consists of modifications applied to the "first-level adaptor", which effectively alter the system's apparent adaptive behaviour.

Let's look now in more detail into the actual operational requirements that the above scenario translates into. In short, the requirements that we will be looking at include: observing interaction; observing adaptive behaviour; self-evaluation; and, modifying adaptive behaviour.

To start with, self-regulation requires that user interaction with the system be observed and interpreted. This requirement should be trivial to satisfy, as this is an integral part of the operation of any (user-) adaptive system. A far less trivial requirement is that the system's adaptive behaviour be observed and modelled. This implies that the system's adaptive behaviour must be "broken down" into relatively discrete constituents (the granularity may vary widely) that can be uniquely identified. As we will see in the next section, although it is possible to relax this requirement somewhat, there are implications which constrain the types of adaptive systems in which self-regulation can be applied / implemented.

The third and, perhaps, most demanding requirement of self-regulation is that of self-evaluation. Self-regulating adaptive systems must be capable of assessing the (degree of) success or failure of the system's adaptive behaviours. Although such assessment may take many forms, this paper will propose and concentrate on an approach that is, arguably, realistic in terms of implementation costs (given a supporting software framework), and whose overhead in terms of adaptation design are not forbidding. This approach is based on the identification of "expectations" in relation to a system's adaptive behaviours, and has been inspired by the work reported in [Browne *et al.*, 1990]. The term "expectations" refers to the anticipated benefits that a particular behaviour will have on the interaction state. Expectations need to be expressed in quantifiable terms and in relation to the adaptive system's dynamic models (which, presumably, comprise a representation of the current interaction state), or to direct user input. The quantified expectations must then be expressed in computable form, and associated with their corresponding adaptive behaviours. Given the proposed approach, self-evaluation can be defined as the process of assessing adaptive behaviours with the computable expectations acting as metrics used to "measure" (degrees of) success or failure.

The fourth and final requirement concerns the capability of adaptive systems to modify their own adaptive behaviour. This implies that the system is capable of either: (a) modifying its first-level adaptation logic at run-time, thus affecting its adaptive behaviour, or (b) leaving the adaptation logic unmodified, but overriding the resulting adaptive behaviours (which, in effect, is equivalent to the establishment of a second-level adaptation logic). Which of the preceding capabilities are plausible for a given adaptive system depends, mainly, on the way in which the system does its decision making. We will return to this topic in the next section.

The above four requirements are, of course, only a sketchy outline of what is needed for self-regulation. Each of the requirements has several additional implications, some of which will be discussed in the next section. Before doing so, though, we need to elaborate on a number of points.

The first such point is how one can address the modeling and quantification of expectations and their fulfillment, in the context of self-evaluation? After all, evaluation of adaptive systems is known to be a problematic issue in its own accord, even when carried out by humans, so shouldn't self-evaluation be next to impossible?

Starting from the second question, let us delineate the most important differences between self-evaluation in the context of self-regulation, and the "external" (empirical) evaluation of adaptive systems. As recent work has shown [Weibelzahl, 2001; Paramythis *et al.*, 2001], because of the inherently complex nature of adaptive systems, identifying the exact reasons for failure of any given adaptive behaviour is quite demanding and requires a structured, methodological approach. Self-evaluation within the self-regulation process, however, need not be concerned with "understanding" *why* an adaptive behaviour fails in a given interaction context, but only that it does – the occurrence of failure can then trigger corrective behaviour on the part of the system. Some may argue that without knowing the reasons of failure, a system cannot possibly hope to provide a viable alternative. As we will see in later sections, for self-regulating systems, this is a point that can be addressed through human intervention. It should be noted, however, that for systems further up the scale of adaptivity (e.g., self-mediating systems), the reasons for failure are almost equally important as the failure itself. In synthesis then, and in the perspective of this paper, self-evaluation of an adaptive system can be informed from, but, at the same time, is entirely distinct from, the "external" evaluation of that same system.

Having established the scope of self-evaluation, let's return to the question of how one models or quantifies expectations. The simplest case, in this respect, would be adaptations that are expected to result in individually observable user actions (e.g., user follows a link specifically annotated to encourage selection). Expectations, in this case, could then be codified as the requirement that such actions occur within given temporal or other constraints. This, in turn, necessitates the presence of: (a) "primitives" which can be used to refer to user actions at a semantic level of abstraction, and (b) "constraint languages" that can be used to apply constraints on the aforementioned primitives. For example, consider an adaptive system in which a set of links are reordered according to a specific adaptation strategy. The expectation to be expressed might then be that users select items (primitive action) from the top of the reordered set (first constraint), soon after the reordering has occurred (second constraint). For exemplification, counter-evidence for the success of the adaptation might be that the users do not select any of the items; or, even worse, select items away from the top of the set.

On the other end of the complexity spectrum would be expectations that can be only approximately expressed, include uncertainty, and require an understanding of the user's interactive behaviour (as opposed to mere observation of the user's actions). Whereas approximate goal descriptions and uncertainty can be tackled through the employment of appropriate reasoning techniques, acquiring an understanding of the user's behaviour is a more involved matter. A pragmatic approach to this requirement would be to express expectations in relation not only to user actions, but also to (changes in) the dynamic models maintained by the system. The premise of this approach is that these models actually embody the system's continuously updated understanding of the "outside world". It is argued that this kind of extension would require few changes in the "constraint languages" discussed above. It would, nevertheless, require a different set of "primitives" capable of capturing the notion of changes in the models, in relation to other dimensions of adaptation (with the temporal dimension playing again a very significant role). To exemplify the concepts discussed, consider the case of an Intelligent Tutoring System, which detects that a user has very limited knowledge of a topic that is a prerequisite for other topics in a delivered course. Adaptations performed at such a stage would be expected to result in the user's knowledge of the topic (primitive) as reflected in the corresponding student model, to be increased (primitive-specific constraint).

A related issue, as mentioned earlier, is that "expectations" need to be associated with specific system behaviours; this, however, is only the design-time part of the picture. At run-time, "expectations" also need to be aware of the context (i.e., the system's current beliefs about the user and the environment, as expressed in the system's dynamic models) within which behaviours were decided upon. This is necessary if the system is to be able to differentiate between contexts in which particular adaptation behaviours have the desired effects, and contexts in which they don't. It is also necessary in order to express "expectations" based on changes in the context over time (e.g., measuring changes in a user characteristic in the system's user model, from the time that a particular adaptation was effected).

A final point to be addressed before continuing to the next section regards what can be considered a marginal case of self-regulating adaptive systems. This is the case whereby the system exposes in some form its adaptation model to the user (much in the same way that user models are exposed in traditional adaptive systems) and allows the user to provide direct and explicit feedback with respect to any specific adaptive behaviour. This would eliminate the need for self-evaluation per se (as this is delegated to the user), but would introduce a host of other problems, not the least among which are the enormous overhead imposed on the user, and the challenge of providing a concise yet meaningful representation of the adaptation model for non-expert users in the first place. Given the above constraints, this approach should be considered unfeasible at the current stage of evolution in the field of adaptive systems.

## 2.2 Implications on Modelling- and Decision making- Approaches

Let's turn our attention now to the implications of the operational requirements posed in the previous section, on the way in which adaptive systems model users (and the interaction context more generally) and decide upon adaptations. We will start from the additional input required for the second-level adaptor.

**Observation**

As already mentioned, self-regulation requires that the system's adaptive behaviour can be observed (and possibly also interpreted). But what exactly *is* observed in this case? The field of adaptive systems is infamous for its lack of standards, or even commonly accepted approaches in this respect. Instead of going into details, which are inevitably bound to specific platforms or architectures (consider, for instance, the differences in adaptive behaviour between hypertext and desktop systems), let us focus on how adaptive behaviour must be manifested in a system, so that it can be observed:

- Firstly, it must be possible for the second-level adaptor to "learn" when the system's adaptive behaviour changes.
- Secondly, it is necessary that the changes incurred be semantically interpretable (i.e., the adaptor must be capable of "understanding" what has changed).

The first of the above characteristics is obviously vital to the operation of self-regulation. The second characteristic, though, is less fundamental than one might originally consider; we will return to this topic shortly. For the time being, it suffices to note that the level and granularity of the interpretation of changes in a system's adaptive behaviour can vary widely. Apparently, the more fine-grained an understanding attainable, the more detailed self-evaluation and subsequent interventions can be.

**Self-evaluation**

After collecting its input, the second-level adaptor proceeds to the stage of self-evaluation. As already mentioned, this stage involves the assessment of the (degree of) success or failure of the system's adaptive behaviour. This implies the capability on the part of the system to quantify the changes that have occurred in the interaction state as a result of applied adaptations. To facilitate discussion, we will assume that such quantification is done through "functions" applied within the second-level adaptor and we will set out to explore their characteristics:

- *Function inputs*: This may comprise direct user input, current values from the static and dynamic models of the system, "historical" values from the same models, as well as interim results from previous calculations. The term "historical" is used in this context to refer to the values that modelled characteristics had at a given point in time, and they are important when comparisons need to be made, to establish the changes in the models brought about through adaptation. This kind of "memory" is not a typical capability of current modelling components, and may need to be provided by the second-level adaptor itself.
- *Function output*: The value domain of the functions' output can be practically anything (e.g., Boolean, discrete, fuzzy, etc.) and actually depends on the computational approach employed for the implementation of the functions (e.g., a probabilistic approach will have a value domain of [0 .. 1]).
- *Computational approach*: This can be the same as the approach used to implement the first-level adaptation logic, but can also be entirely different. Elaborating on the last point, it is interesting to note that, given a sufficient degree of similarity in how adaptive systems communicate model and adaptation information to the second-level adaptor, it is possible to create a generic computational approach to self-evaluation that can be used "above" several different types of first-level adaptation logic.
- *Association with the adaptation(s) being evaluated*: This has already been briefly discussed in the previous section, and an assertion was made that it is not as fundamental as one might expect at first sight. As we will see in the next section, for some types of intervention (i.e., modification of the first-level adaptive behaviour), the only thing necessary is that adaptations can be uniquely identified, so that the two adaptors can "converse" about them (e.g., the second-level adaptor instructing that a specific adaptation be disallowed for the current user). As we will see, even the requirement for *unique* identification can be relaxed, if interventions occur at a sufficiently low level – however, adaptation types would still need to be identifiable.

**Modifying the system's adaptive behaviour**

Following self-evaluation, the second-level adaptor may need to intervene and modify the first-level adaptive behaviour. This implies that the system is capable of either: (a) modifying adaptation logic at run-time, thus affecting its adaptive behaviour, or (b) leaving the adaptation logic unmodified, but overriding the resulting adaptive behaviours. The run-time modification of adaptation logic is a process that is evidently dependent on the type of logic used (e.g., in rule-based systems, it would signify modification of rules; in systems based on Bayesian networks, it would signify changes in the network, etc.). As such, this type of intervention is too complex and broad a subject to address this paper. Instead, we will focus the second method of intervention, which does not presuppose any modifications to the first-level adaptation logic.

This second method requires that the second-level adaptor can override adaptations decided upon by the

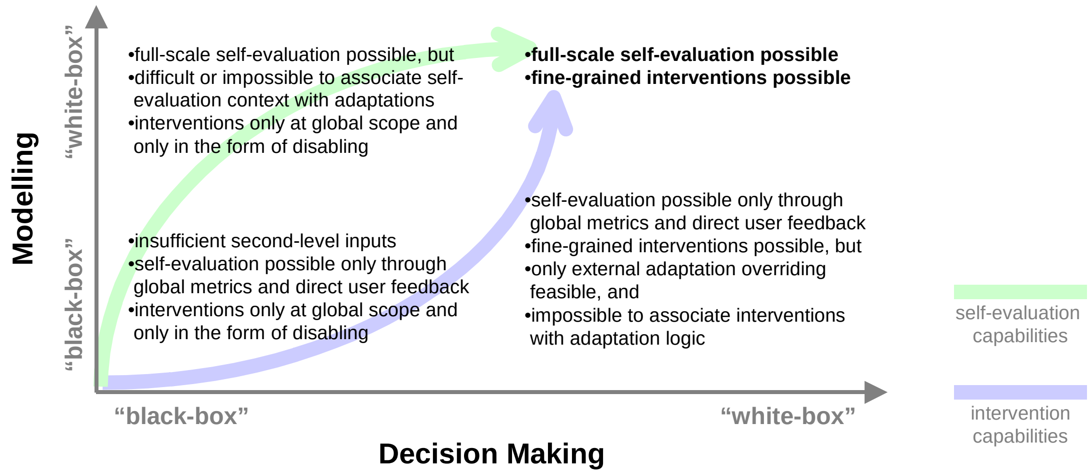


**Figure 2:** The effects of the exposed granularity of the modelling- and decision making- models on an adaptive system's self-regulation capabilities.

first-level one. Overriding, in this context, can take several forms, the most important among which are:

- *Disabling adaptations*: Arguably the simplest form of overriding is to disallow adaptations from occurring when there is evidence that they have detrimental effects on the interaction. If adaptations can be uniquely identified and associated with a context (i.e., user characteristics, interaction state, etc.), then disabling can occur at a quite fine-grained level. Lack of unique identification, and, similarly, lack of context associations, results inevitably in more "global" overriding effects (i.e., *all* instances of a particular adaptation type are disallowed, or a specific adaptation is disallowed in *all* contexts; apparently, this may result in disabling by implication even adaptations that had positive effects on the interaction).
- *Constraining adaptations with weighting functions*: This form of overriding is based on the employment of additional functions that use the results of self-evaluation to "promote", or, more usually, "demote" adaptations. Demotion, in this context, refers to the application of additional constraints on the circumstances under which an adaptation is allowed to take place (with promotion having the converse effect). These constraints might be entirely independent from those used for deciding upon the adaptation at the first level. One plausible approach to such weighting functions would be, for instance, "utility" functions, as described in [Horvitz, 1999], or, from a different perspective, in [Herder, 2003]. Disabling adaptations, as discussed above, can be seen as a special case of constraining, with a Boolean weighting function.
- *Using alternatives*: This form of overriding presupposes the presence of alternatives for given (types of) adaptations. Note that the second-level adaptor does not necessarily need to understand the differences between alternatives. Using a trial-and-error approach, for example, would enable the adaptor to identify the one most suitable for a given context, without knowing how the alternatives actually differ. Disabling adaptations can also be seen as a special case of using alternatives, with two alternatives for each adaptation, one being the "null" or "empty" alternative. Although promising, this approach incurs additional overhead in the design and development of the adaptive system, as the adaptation model will need to have a representation of the alternatives themselves and of their associations with logic (see, e.g., [Savidis *et al.*, 1997]).
- *"Editing" adaptations*: The most sophisticated form of overriding is to actually modify the adaptation itself. This is a quite demanding endeavour, as it requires that the second-level adaptor: can acquire semantic information about an adaptation's constituent parts[2] and their individual effects on interaction; can modify these constituents to achieve different effects by altering their parameters, or by replacing them with alternatives, or, even, by simply removing them. Arguably, the most challenging part in all this is that the above process presupposes the existence of meta-knowledge that would allow the second-level adaptor to decide what exactly to change and why. A complete solution in this respect falls more within the scope of self-mediating systems, rather than self-regulating ones. A pragmatic approach, however, may be based on the concept of "templates" which could specify ways in which the adaptor can modify specific categories of adaptations. The task then would be reduced to identifying (on the basis of the self-evaluation results, the context associated with the adaptation, and on the nature of the adaptation itself) which template needs to be applied.

The above enumeration of possible forms of intervention is, of course, not exhaustive. Furthermore, the forms discussed are by no means mutually exclusive – although

[2] For example, one way of decomposing adaptations in this manner is to break them down to primitive adaptation actions (see, e.g., [Paramythis and Stephanidis, in press]).

it is unlikely self-regulating systems will support all of them simultaneously. Apart from the overhead involved in designing and developing increasingly sophisticated interventions, there is the fundamental question of what can be achieved, given an existing adaptive system.

**Implications**

It may seem that given the proliferation of a wide range of modelling- and decision making- approaches in use today, and the fundamental differences between them, the preceding question can only be answered on a per-case basis. It is argued, however, that, at a high level of abstraction, there is one dimension that is by far the most important with respect to self-regulation: the level and granularity at which the internals the modelling- and decision making- processes are exposed to the rest of the system. We will borrow the terms "white-box" and "black-box" to refer, respectively, to the case of the process internals being fully inspectable by the rest of the system, and the case of having no possibility for inspection at all. Further, "black-box" and "white-box" are to be understood as two fictional endpoints of a continuum, with increasing levels of inspectability and granularity leading from one to the other.

The factor most likely to determine where in this continuum a particular modelling- or decision making- approach belongs is the computational character of the algorithms that implement it. For instance, consider the case of a system that uses neural networks to associate dynamic model attributes with adaptive behaviours. Since the "internals" of the neural network do not have individual semantic value, even if they were to be exposed they would be of no use to the rest of the system. That system's decision-making approach would then lie at the "black-box" end of the spectrum. Conversely, consider a system that uses rule-based adaptation logic. As rules are distinct and, at least theoretically, possible to manipulate individually, they would result in the system's being classified as having a "white-box" decision making approach.

Figure 2 presents a high-level overview of how the inspectability and granularity of the modelling- and decision making- processes affect the self-regulation capabilities of an adaptive system. Although a comprehensive discussion is beyond the scope of this paper, a few notes are in order:

- Self-evaluation capabilities are mainly dependent on the inspectability and granularity of a system's modelling approach.
- Intervention capabilities are mainly dependent on the inspectability and granularity of a system's decision making approach.
- Implementing self-regulation in systems with "black-box" modelling would effectively necessitate additional modelling performed at the second level adaptor, on the basis of direct user input.
- Implementing self-regulation with "black-box" decision making would require that the second level adaptor can "reverse" (or otherwise modify) the *potential effects* of adaptations, as the adaptations themselves are opaque.
- Implementing self-regulation in a model- and logic-agnostic manner is still possible, but requires that the second level adaptor: (a) can directly interpret direct user input; (b) supports a concept of context, based on that input; (c) supports a generic concept of self-evaluation along the same lines; and (d) applies second-level adaptations independently of the first-level adaptor.

## 3 Discussion

The previous sections have attempted to provide an overview of how self-regulation can be understood in the context of, as well as of the prerequisites it imposes on, modern adaptive systems. One of the several important topics that have not been discussed thus far, is how the system can be sure that the changes it observes on the interaction are attributable to a specific adaptation, while performing self-evaluation? This is a fundamental question to which the answer is, perhaps unsurprisingly: it can't! At its core, self-regulation is restricted to assessing whether system behaviours have the expected results, but there are only "extrinsic" ways for the system to ensure that these results were not side effects of other, entirely unrelated behaviours.

One such way is the establishment of "clusters" of collaborating self-regulating systems. Clustering, here, refers to the establishment of communication and coordination channels between the systems. The subject of collaboration is none other than the systems' aggregated "findings" in the second-level adaptation cycle. For instance, an adaptive learning system could communicate to its cluster that it has observed that a particular type of adaptation (e.g., link annotation) has not had the expected results (e.g., the user did not choose the annotated links over non-annotated ones) in some of the hosted courses; furthermore, the system could attach to that observation the attributes of the models describing the interaction state that were common in all these observations (e.g., that the user has considerable computer expertise, and that the user has prior knowledge of the subject domain). Other systems in the cluster could then refine (or challenge) the asserted observation, with their own. It is argued that the power of this approach lies with the fact that the validation of the first-level adaptive behaviour happens in large scale, and is based on the statistical validity of contributed observations. Please note that, although the preceding example is of a negative observation, positive observations would not only be equally interesting, but also vital to the operation of the cluster.

This form of "sharing of experience" within clusters of self-regulating systems can result in a body of meta-knowledge regarding the systems' basic, or first-level adaptive behaviour. This meta-knowledge is of value unto itself, as it would, in several cases, suffice to answer questions such as the one posed at the beginning of this section. It would also make it possible for newly-installed same-domain systems to take advantage of the accumulated "experience" of other peer systems, as this is expressed at the level of the cluster. Furthermore, creating clusters that support exchanges between systems operating within different contexts of use would enable us to derive more generalised knowledge, and perhaps even "discover" cross-application or cross-domain interaction patterns with relevance to adaptation. Finally, the knowledge accumulated within clusters could serve as the basis for adaptation models that would enable the development of grounded self-mediating systems.

Another important topic that merits our attention is the redefinition of the role of adaptation designers, as well as the more general role of human experts in relation to self-regulating systems. To start with, self-regulation demands that we revise the way in which we design adaptive systems. To date, the knowledge and rationale behind adaptation design may exist (although the literature indicates that sometimes common sense and intuition are the sole basis of designs), but is definitely not "codified" into the adaptive system. As discussed earlier, this knowledge now needs to be formalised and expressed as measurable "expectations" that the system assesses against. It is argued that although this requirement may imply additional overhead in the design of adaptive systems, it also has the potential of improving designs in the first place.

Apart from changing design practices, self-regulation calls upon human experts to undertake new roles in the adaptation process. Specifically, humans may now need to (occasionally) inspect the modifications effected on the system's behaviour by the second-level adaptor and intervene when the system is evidently at fault. More interestingly, humans are called upon to semantically interpret the "findings" of self-regulating systems working in isolation or within clusters, or resolve conflicts in the latter case. Finally, the aforementioned "findings" have the potential to inform, or even serve as input, to the empirical evaluation of adaptive systems, which, in turn, can help improve both the first- and second- level adaptive behaviour.

There are several questions of pivotal nature in the theoretical and practical employment of self-regulation that have not been addressed in this paper. For instance, what is the role of the adaptive system's goal against which the evaluation should take place? Should the goal be fixed and concrete, or should the self-regulating adaptive system be able to deal with less concrete goals or goals that change over time? Are there clear limits between self-regulation and self-mediation? Although the space available was not sufficient to cover them, these and other questions need to be brought to the epicenter of discussion, as they are at least equally important for the adoption of self-regulation as the more "technical" issues discussed herein.

The final topic that we would like to touch upon in closing is the steps we need to make as a community to move closer to the establishment of self-regulation as a standard property of adaptive systems. It is argued that realistic path will go through the following milestones: (a) establishment of a comprehensive theoretical basis for self-regulation; (b) development of software frameworks that can provide basic self-regulation capabilities as an "add on" to existing adaptive systems; (c) experimentation and validation on systems with "white-box" modelling- and decision making- approaches; (d) accumulation and synthesis of experiences, towards a more broad dissemination of the involved theory and technologies.

## Acknowledgements

The work reported in this paper has been supported by the project "Integrating Agents into Teleteaching Web-portals" sponsored by the Austrian FWF (Fund for the Support of Scientific Research; project P15947-N04).

## References

[Benyon, 1993] Benyon, D. Adaptive Systems: {A} Solution to Usability Problems. *User Modeling and User-Adapted Interaction*, 3(1): 65—87, 1993.

[Browne *et al.*, 1990] Browne, D. P., Norman, N. and Riches, D. Why Build Adaptive Systems? In Browne, D., Totterdell, P., and Norman, M. (eds.): *Adaptive User Interfaces*, pages .15-57. *Adaptive User Interfaces*. Academic Press, London, 1990.

[Herder, 2003] Herder, E. Utility-Based Evaluation of Adaptive Systems. In Weibelzahl, S. and Paramythis, A. (eds.), *Proceedings of the Second Workshop on Empirical Evaluation of Adaptive Systems*, held at the 9th International Conference on User Modeling UM2003, pages 25-30, Pittsburgh, USA, 2003.

[Horvitz, 1999] Horvitz, E. Principles of Mixed-Initiative User Interfaces. In *Proceedings of CHI '99, ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 159-166, Pittsburgh, PA, USA, May 1999. ACM Press, New York, 1999.

[Paramythis *et al.*, 2001] Paramythis, A., Totter, A. and Stephanidis, C. A Modular Approach to the Evaluation of Adaptive User Interfaces. In Stephan Weibelzahl, David Chin, and Gehard Weber (eds.) *Empirical Evaluation of Adaptive Systems*. Proceedings of workshop held at the Eighth International Conference on User Modeling, pages 9-24, Sonthofen, Germany, July 2001. Pedagogical University of Freiburg, Freiburg, 2001.

[Paramythis and Stephanidis, in press] Paramythis, A., and Stephanidis, C. A Generic Adaptation Framework for Hypermedia Systems. In Chen, S. Y., & Magoulas, G. D. (eds.) *Adaptable and Adaptive Hypermedia Systems*. Idea Group, Inc. , in press.

[Savidis *et al.*, 1997] Savidis, A., Paramythis, A., Akoumianakis, D., and Stephanidis, C. Designing user-adapted interfaces: the unified design method for transformable interactions. In *Proceedings of the ACM Conference on Designing Interactive Systems: Processes, Methods and Techniques (DIS '97)*, pages 323-334, Amsterdam, The Netherlands, August 1997. ACM Press, New York, 1997.

[Totterdell and Rautenbach, 1990] Totterdell, P., and Rautenbach. Adaptation as a Problem of Design. In Browne, D., Totterdell, P., and Norman, M. (eds.): *Adaptive User Interfaces*, pages 61-84. Academic Press, 1990.

[Trevellyan and Browne, 1986] Trevellyan, R., and Browne, D. P. A self-regulating adaptive system. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*, pages 103-107, Toronto, Ontario, Canada, 1986.

[Weibelzahl, 2001] Weibelzahl, S. Evaluation of adaptive systems. In M. Bauer, P. Gmytrasiewicz, and J. Vassileva (eds.), *User Modeling 2001: Proceedings of the Eighth International Conference, UM2001*, pages 292-294 (Lecture Notes in Computer Science). Springer, Berlin, 2001.