Shared initiative: Cross-fertilisation between system adaptivity and adaptability

Markus Klann, Markus Eisenhauer, Reinhard Oppermann, Volker Wulf

Fraunhofer Institute for Applied Information Technology FIT
Fraunhofer FIT, Schloss Birlinghoven, 53754 Sankt Augustin
{markus.klann, markus.eisenhauer, reinhard.oppermann, volker.wulf}@fit.fraunhofer.de

Abstract

In the present article we investigate a new way of how computer systems can better meet their users' requirements. We start from the well-known notions of situation-aware adaptivity, automatically carried out by the system, and adaptations, consciously carried out by the users. We indicate the shortcomings of both of these approaches and show how they can be compensated for, at least partially, by the respective other approach. We argue that such a shared initiative of both system and user adaptations, mutually supporting each other, provides a considerable advantage in keeping a computer system in line with dynamically changing user-requirements.

1 Introduction: the concept of shared initiative between system and user control

Today, software systems are faced with the problem of catering to the diverse and changing requirements of heterogeneous groups of users. Cutting down the requirements to a supposedly fitting least common denominator for *all* users means not taking up the challenge. The solution must be to master the heterogeneity of requirements and provide means to handle it. In this article, we investigate how software systems can reach a better fit with the diverse requirements of their users, by considering the joint benefits of two different approaches[1]: First, situation-aware adaptivity, meaning that the system adapts automatically to its users according to the situational context. Second, adaptability, meaning that the users themselves can substantially customize the system through tailoring activities.

Both of these approaches keep the system flexible during usage. Such a flexible system, which adapts to its users and which the users can adapt according to their needs and preferences, should be easier to handle and should enhance the users' productivity, optimize work-loads, and increase user satisfaction.

As it is impossible to anticipate the requirements of all users, a single best or optimal system configuration is impossible. Therefore, the task is to find a suitable trade-off between automatic system adaptivity and user controlled adaptability, resulting in a flexible system through shared initiative. We expect that system flexibility can be enhanced by exploiting situation-aware adaptivity for the system's tailorability, and vice versa.

[1] Of course, there are a large number of other approaches to capturing user requirements, notably during the design phase of computer systems (e.g. participatory and scenario-based design, visual and domain-specific languages for users to express their requirements).

2 State of the art: adaptivity and adaptability

Both adaptivity and adaptability of computer systems have received much attention in research during recent years (see (Oppermann, 1994) and (Krogsæter, 1994)). But research has addressed these two aspects largely independently of one another. The question of how these two aspects might benefit from each other has not yet been thoroughly investigated.

2.1 Adaptivity

The aim of adaptivity is to have systems that adapt themselves to the context of use with respect to their functionality, content selection, content presentation and user interactions. Systems displaying such adaptive behaviour with respect to the context of use are called situation-aware.

One aspect of situation-awareness is related to properties of the user itself, like the level of qualification, current task or previous behaviour. Traditionally, these properties have been captured in user models, which have been processed to generate appropriate adaptive behaviour. Currently, situation-awareness is continuously augmented by taking more and more situational properties into account. In particular, various sensors are used to gather information on properties relating to the physical environment of the context of use, like the time of day, location, line of sight, level of noise, etc. Other situational properties of the context of use of a particular user relate to what may be called the social environment, being composed of other users, communicative and cooperative interactions, shared artefacts and common tasks. One example of such an adaptive system would be a tourist information system on a mobile computer that presents specific informations based on its user's location, movement, profile of interests etc.

The basis for a successful and effective information and communication system is providing information and functionality that is relevant and at the right level of complexity with respect to the users' changing needs. As these changing needs are largely related to the situational properties, relevance and appropriate complexity can be supported by system adaptivity, which is to say by automatic proactive selection and context-sensitive presentation of functionalities and contents.

The objective is to assist the users by proactively supplying what they actually need. This way, users are not distracted from their primary task by searching and selecting. A good quality of such adaptivity clearly depends on complete and accurate user- and context-models, as well as on correct conclusions derived from them.

2.2 Adaptability

The aim of adaptability is to empower end-users without or with limited programming skills to customize or tailor computer systems according to their individual, context- or situation-specific requirements. Approaches to adaptability include:

- End-user-friendly programming languages, see e.g. (Repenning, 2000)
- 'Programming by example'[2] respectively 'Programming by demonstration', see (Cypher, 1993) and (Lieberman, 2001)
- Component-based tailoring, see e.g. (Stiemerling et al., 1999)

By avoiding costly and time-consuming development cycles with software engineers whenever possible, such approaches allow for fast adaptations to dynamically changing requirements by letting the end-users put their domain specific expertise to the task of system customization. While currently

[2] Taken as a conscious activity and not as an accidental side-effect to usage, ‚Programming by example' constitutes a case of adaptability. But as it also requires system activity, namely deducing some function from the user's behaviour, it may also be considered a case of adaptivity.

still being in its infancy with simple adaptations like macros for word processors or email filters, more sophisticated forms of adaptability should enable end-users to become the initiators of a co-evolution between the systems they are using and their own requirements as defined by their tasks, level of expertise and current working context.

3 How to enhance adaptivity through adaptability

Automatic system adaptations are only possible to the extent that the system can gather the required information for performing some adaptation function. And they are only reasonable to the extent that the system can assess which adaptation functions are suitable for the current context and in case of multiple adaptation options which one to carry out with sufficient benefit.

In some cases, sufficient information can be gathered through sensors or other sources of information, like user preferences or interaction histories to derive useful adaptations and carry them out. But in other cases the required information for some adaptation is not available or it is not possible to select an appropriate one among a set of potential candidates.

In both cases, adding adaptability can help to still benefit from adaptive system functionality. Obviously, this is done by either letting users provide missing information in order for the system to automatically carry out some adaptation function or by having the user make the selection if multiple options exist. In the example of the mobile tourist information system this could mean that the system has tracked previous user activities as a basis for making recommendations that match the user's interests. The user might adapt this functionality by either adding or correcting information or by modifying the strategy by which the system acquires information and arrives at its conclusions.

These user-adaptations can of course be very simple. They might be nothing more than providing missing parameters or actually choosing among a list of adaptation options. But they might as well be rather complex. They might consist of defining what situational properties are to be taken into account by some adaptivity function as relevant for the current context . Or they might consist of defining an actual compound function that implements an adaptivity strategy suitable for the current situation. In both cases, users can exploit their superior awareness of their situation and their domain expertise to enable system adaptivity that would otherwise not be possible.

4 How to enhance adaptability through adaptivity

As any other system functionality, adaptability can benefit from adaptivity. As seen above, user adaptations can be very complex and correspondingly difficult and cognitively demanding. System adaptivity can help reduce the cognitive load on the customizing end-user by hiding those adaptation functions that are not pertinent in the current context.

Moreover, a system might provide adaptability at different levels of complexity, geared towards adaptations of varying degrees of difficulty. An adaptive system could choose an appropriate level of tailoring complexity based on the current task and level of tailoring expertise of its user.

Alternatively, a situation-aware system might be able to identify recurring tailoring situations based on the current task, pursuit goal, involved people etc. and might be able to suggest the reuse of existing tailored artefacts to its users or to suggest getting assistance from other users who have successfully tailored in similar situations. Thus, situation-aware systems might be very important to foster collaborative tailoring and the sharing of tailored artefacts (Wulf, 1999). As an example, a groupware system might present only those configuration options to its users that are appropriate for their level of expertise and their mutual task.

Finally, system adaptivity is actually of particular importance to adaptability for a specific reason. This is because adaptability functions should be as unobtrusive as possible on the user interface during normal use (Wulf, 2001): if the user feels no need to tailor, as little as possible of his attention should be deviated by tailoring functionality. This can be achieved by adaptively displaying only those tailoring functions on the user interface that are likely to be of use, based on such properties as current task, level of expertise, available time etc. Thus, adaptivity can play a crucial role in reducing the cognitive load of tailoring functionality and consequently raising the users' inclination to carry out tailoring activities.

5 Discussion of shared initiative: the joint benefit of adaptivity and adaptability

As explained in the preceding sections, automatic situation-aware adaptivity and manual adaptability are important features of computer systems that are supposed to sustainably cater to their users' needs. However, it was also shown that both of these features have their specific limitiations and disadvantages.

However, it was also shown that both means, the automatic situation aware adaptivity and the manual adaptability, have their specific limitations and disadvantages. In particular, the first is incomplete and imprecise and reduces the users to a passive receptionist of automatic mechanisms. The latter is costly, requiring additional effort from the users for the meta-task of tailoring the system. Nonetheless, we believe that end-user tailoring is necessary, as today's high degree of situation-aware system adaptivity does not allow for software designers to anticipate all possible ways of system behaviour. On the other hand, today's higher degree of situation-awareness may allow for better proposals for tailoring activities.

A combination of both adaptivity and adaptability may help to overcome their individual problems by preparing a spectrum of increasingly conservative best guess proposals. The best guess proposals consider the current users, task and environment and they are presented as a zoomable spectrum with appropriate content and suitable interaction primitives. Such best guess solutions empower the user itself to comfortably browse from the most specific best guess option to alternative options requiring an increasing amount of user adaptations.

6 Conclusion

In this article we have shown how the shared initiative of adaptation activities, automatically carried out by an adaptive situation-aware computer system and consciously carried out by its user may lead to a synergetic advantage with respect to a continuous close fit between the system and its user in the presence of dynamically changing requirements.

The envisioned interplay of adaptivity and adaptability obviously constitutes a new kind of system behaviour. For users to fully exploit the potential advantages of such a shared initiative, that is, to rely on system adaptivity, to confidently carry out adaptations and to benefit from their interplay, users will probably have to change their expectations on how such computer systems operate.

How this change of expectations can be facilitated and supported, as for example by visualizing dependencies and the consequences of operations or by pedagogical efforts is a question for future research.

References

Cypher, A. (1993). *Watch What I Do: Programming by Demonstration*. Cambridge, Massachusetts: The MIT Press

Krogsæter, M., Oppermann R., et al. (1994). A User Interface Integrating Adaptability and Adaptivity. In R. Oppermann (Ed.), *Adaptive User Support. Ergonomic Design of Manually and Automatically Adaptable Software* (pp. 97-125). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Lieberman, H. (2001). *Your Wish is My Command: Programming By Example*. San Francisco: Morgan Kaufmann

Oppermann, R. (1994). *Adaptive User Support. Ergonomic Design of Manually and Automatically Adaptable Software*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Repenning, A., Ioannidou, A., & Zola, J. (2000). AgentSheets: End-User Programmable Simulations. *Journal of Artificial Societies and Social Simulation*, 3 (3).

Stiemerling, O., Hinken, R., & Cremers, A. B. (1999). The EVOLVE tailoring platform: supporting the evolution of component-based groupware. In IEEE Communications Society (Ed.), *Proceedings of the Third International Enterprise Distributed Object Computing Conference, EDOC '99 : 27 - 30 September 1999, University of Mannheim, Germany . Mannheim* (pp. 106-115). Los Alamitos, Calif.: IEEE Press.

Wulf, V. (1999). "Let's See Your Search-Tool!" - On the Collaborative Use of Tailored Artifacts. In S. C. Hayne (Ed.), *Proceedings of the international ACM SIGGROUP conference on Supporting group work, GROUP '99* (pp. 50-60). New York: ACM Press.

Wulf, V., & Golombek B. (2001). Direct Activation: A Concept to Encourage Tailoring Activities. *Behaviour and Information Technology*, 20 (4), 249-263.