Design Principles for Embodied Interaction: The Case of Ubiquitous Computing

Rainer Malaka and Robert Porzel

University of Bremen,
TZI, Digital Media Group
Bremen, Germany
{last name @tzi.de

Abstract. Designing user interfaces for ubiquitous computing applications is a challenging task. In this paper we discuss how to build intelligent interfaces. The foundations are usability principles that are valid on very general levels. We present a number of established methods for the design process that can help to meet these principle requirements. In particular participatory and iterative so-called human centered approaches are important for interfaces in ubiquitous computing. In particular the question how to make interactional interfaces more intelligent is not trivial and there are multiple approaches to enhance either the intelligence of the system or that of the user. Novel interface approaches, presented herein, follow the idea of embodied interaction and put particular emphasis on the situated use of a system and the mental models humans develop in context.

1 Introduction

It is well known that user interfaces for computational devices can constitute a challenging matter for both their users and their designers [1, 15]. By definition an interface is at the boundary of two entities, which - in the case of user interfaces lies between humans and machines while other interfaces lie, for instance, between networks and computers. In ubiquitous and pervasive computing scenarios [33], we face the problem, that there might be no clear boundaries any more. Computers are truly no longer visible at the end and they can disappear from the user's conscious perception. We will, therefore, face the challenge to build an interface for something that is rather shapeless and invisible. Thereby traditional means of employing affordances [20] may no longer apply and the critical notions of context-awareness and situated interaction come into play.

Generally speaking, new forms of enabling meaningful and felicitous interactions in dynamically changing contexts have to be found. A specific proposal for an alternative form of interaction - motivated by the shortcomings of so-called *representational* approaches for including context - has been put forward under the heading of *interactional approaches* to context [7]. In short, this approach - situated itself in the embodied interaction framework [4] - seeks to employ the interface to enable the user to make intelligent decisions, rather than relying on the system to make a decision. This particular way of fleshing out the principles of embodied interaction, however, is thwarted in the ubiquitous computing scenario introduced above where that option is no longer viable or - in a manner of speaking - has disappeared together with the interface itself.

In this work, we will propose an alternative solution that provides an extendable combination of representational and interactional instruments in an embodied interaction approach viable for everyday computing. While prototype implementations exists [18] together with a range of prior work and experimental studies, the focus of this work is not to describe a specific technical realization or evaluation thereof, but to flesh out the principles for designing computational blueprints for approaches that are based on the awareness of the shortcomings of prior approaches as well as on the contributions that come from ethnomethodological and psychological considerations [27, 9].

In the following, we will go in more detail through these principles and will introduce some general approaches for designing user interfaces. We will see that we one learn from interface design for other - classical - scenarios, and still apply many of those user interface design principles for ubiquitous computing as well. A central aspect thereby is design process that helps to find the right sequence of steps in building a good user interface. After discussing these general aspects of user interface design we will focus on the specific needs for fleshing out embodied interaction in ubiquitous computing scenarios and finally on how to build intelligent user interfaces - or to put it the other way around - to circumvent interfaces that do not facilitate efficient and felicitous interaction.

2 Prior Art

The design of a good user interface is an art, which has been ignored for a long time in the information and communication business. Frequently developers implemented what they thought useful and assumed it to be beneficial for the respective users. However, most users are not software developers and their way of interacting with technology can differ radically. The result is technology that despite its functionality is intuitively usable only for a small group of people. For others it can be highly inefficient, frustrating or even unusable. Some of the highlights of this dilemma can be found in the communication with the user when something goes wrong. Error messages notifying the user often are of use for the developer, e.g. to help in his efforts in debugging the system, but frequently a user will not be able to understand what happened.

In contrast, today usability plays an important role and many systems are designed with more consideration for intuitive and (fail)safe usage. On the one hand this is due to obligations concerning accessibility, but also due to the fact that many systems do not differ greatly in functionality and technical detail and vendors have to diversify their products mainly in terms of their *look and feel*. There exists now a wealth of empirical methods, tools, and guidelines, which all help to develop good user interfaces [28, 25, 6]. However, there is not one single recipe whose application guarantees perfect results. One essence of usability engineering - in all cases - is to work iteratively in order to achieve the goal of better usability.

Consequently, much prior work on iterative and agile development still applies for designing intelligent user interfaces for ubiquitous computing, while they are also valid for other user interfaces such as *visible* web- or desktop interfaces. The general process of human-centered design could even be applied for products such as machines, appliances and other artifacts [19]. As a matter of fact, one can see this as a truly generic process. With this perspective good usability is a property that is generic featuring similar design processes for multiple domains which aligns perfectly with the way in which ubiquitous computing seeks to integrate everyday computing into the objects of our normal life. From this point of view, then, usability engineering and ubiquitous computing are both concerned with the usability of everyday things, whether they be soft (digital) or hard (physical) components of the intertwined ecosystem.

As a result, from its very beginning on ubiquitous computing had usability in its focus. Mark Weiser's idea of ubiquitous computing encompasses *invisible* interfaces that are so naturally usable that they literally become invisible for the users conscious perception [34]. This notion is expressed by the philosophers Martin Heidegger and hos student Georg Gadamer who called such an interaction with things that we use without conscious awareness things that are *ready-to-hand* or at our *horizon* [13]. In this phenomenologist view, the meaning of the things is actually derived from our interaction with them. Such a view on interactive artifacts can be adopted in ubiquitous computing and is closely related to the notion of embodiment [5, 7, 12]. This is a fundamental shift from the classical positivist approach that was taken in computational systems. Specifically, this shift goes from modeling the real world via simplifying abstractions to an embodied approach that puts the users and their context at the heart of the matter.

The shift from classical positivist approaches to interactional phenomenologistic approaches is pertinent for finding suitable new forms of interfaces for ubiquitous computing. This is case on the one hand if applications of ubiquitous computing are to be used in *rich* dynamic real-world settings - as opposed to *poorer* closed world settings which feature one or a specific number of contexts and all other contexts have been abstracted away, usually in an implicit manner [22] - then, as a consequence, the way in which *meaning* is afforded for the user will, in fact, evolve in the course of action. But also if applications of ubiquitous computing should become natural extensions of our physical abilities, they must be designed such that they do not need more conscious interference from their users than other natural embodied forms of interaction. Please note that this notion of being *invisible* does not necessarily imply *not there*, but rather present without requiring conscious interaction with the form, but rather natural employment of the form as one finds it in language-based interactions among human interlocutors.

The common examples for interacting with physical objects employ our body parts. When we grasp a coffee cup, we just take it and we do not think and plan how to make our hand grasp the handle and to make our arm to bring the hand there. In this sense, the arm and hand are invisible but also very present. Thus, speaking of interfaces for ubiquitous computing as being *invisible* or computers that are *disappearing* we actually speak of things that are present and *ready-to-hand*. As a result, ubiquitous computing artifacts that one interacts with might not be visible as computers or components. Humans have economical models how things works that are internalized to such a degree that we do not have to think about them, unless we do so professionally [14]. As stated above, these so-called *mental models* of how things work play an important role in designing good user interfaces as well as in designing other everyday things [19, 20]. Don Norman emphasizes that providing a good design comes down to providing the following mappings in an adequate manner: the *design model* must be mapped to the system image, the users must be able to map their mental model(s) to the system which must allow the user to map its image to the users' models.

The question is now, how can a system image support the appropriate user's mental model. The answer - with the notion of embodiment in mind - must bring the meaning of things into the things themselves - thereby enabling a user to derive the meaning of something from the interaction with it, for example, via its appearance that signals some properties indicating how to use it. This, of course, brings up the central notion of affordance again where the idea of affordances now is to bring knowledge into the world and interface instead of having it in the mind, e.g. remembering a sequence of commands or button presses or what to do with chairs. Many usable things in our environment let us know by their physical appearance how to use them. A chair, for instance, does not need a label, manual or instructions on how to sit on it. We just see and know that it can serve as a chair in this context because we know what we can do with it, enabled via embodied simulations that *run on* the respective mental models.

Consequently, this notion of affordances has been adopted in the form of *virtual* affordances for computer interfaces and numerous metaphors on our computer screens signal functionalities, e.g., mouse pointers and scrollbars. With the advent of ubiquitous computing, an affordance becomes again more literally a property attached to the physical properties of things. Not surprisingly, therefore, many ubiquitous computing objects include tactile interfaces or objects with both physical and virtual properties.

3 Principles of Embodied Interaction for Ubiquitous Computing

There are a number of consequences following from assuming this perspective and line of argumentation for fleshing out embodied interaction principles for ubiquitous computing [15]:

- Support mental models - humans employ mental models to understand and to predict how things react to their actions. The system image should support such mental models and make it therefore understandable and usable;
- Respect cognitive economy - humans re-use their mental models and if well-established mental models for similar things exist then they constitute a good basis for providing a means to understand what a new artifact could afford;
- Make things visible and transparent - in order to understand the state of an object it should be obvious what is going on at the appropriate level of granularity, e.g., both containers and folders can indicate how loaded they are;
- Design for errors - mappings between the users' models and the systems sometimes fail and many *human errors* are, in fact, mapping errors. Therefore, systems must assist users in finding a solution for their task even if something went wrong, for which there are a number of techniques, e.g., allowing undo-actions or checks;
- Internal and external consistency - things within an application should work consistently, for instance, pushing a certain type of button, e.g. a specifically colored one, always carries some consistent signal. As a common example the color red has been conventionalized in our culture for denoting stops, which is also pertinent for external consistency where expectations users may have from usage of other

applications, i.e. if we add some ubiquitous computing technology to an artifact, e.g. when turning a *normal* cup into a smart cup, a user will still expect the artifact to retain its normal function(s), or expect the cup to work also as a cup.

Additionally, for the case of our invisible interface forms, where some set of (inter)actions are afforded by the environment that has been instrumented by means of ubiquitous computing technologies. As specified above this instrumentation should only add new affordances or augment - or extend - already existing affordances without obscuring or disabling them. Situated interaction in such an environment becomes feasible by means of situational awareness and corresponding contextual computing approaches.

It has been pointed out that current implementations of context-dependency or context-awareness in computational systems follow an almost standardized path [7]. Firstly, a set of possible environmental states of contextually relevant parameters are defined; then, rules are implemented that try to match sensory inputs to one of the given states during runtime. Within these types of applications context-awareness is fundamentally provided by such matching processes and context itself is represented by the predefined and stored set of environmental settings.

Hereby it is not only difficult to determining the appropriate settings or states of the pertinent parameters, but also the fundamental problem of this approach to contextual computing hinges of the question of how one can pre-compile all the settings and parameters that may become pertinent in advance. When applications of ubiquitous computing are to be used in the aforementioned *rich* dynamic real-world settings - it becomes impossible to define these settings and parameters based solely on past research, surveys, testing, own experience, and on the purpose of the particular system alone.

Correspondingly, given versatile form instruments, as applied in multimodal systems [32, 16] it becomes virtually impossible to predict all the possible input and the corresponding contextual dependencies on which their interpretation might hinge. But even in seemingly less murky waters human behavior can hardly be predicted as pointed out frequently by the example of cell phone use as watches, alarm clocks, flash-lights or short message terminals. These examples show that people may use and interact with technology in unexpected ways. This reveals a fundamental problem of implementing a predefined set of settings as such approaches will inevitably not scale to cover possible interactions and behavior that will occur or evolve in future. One reason for this problem is that context has been approached as a representational problem by assuming the following properties of context [8]:

- context is a form of information, i.e. context is seen as something that can be known, represented and encoded in software systems;
- context is delineable, i.e. it is thought to be possible to define what counts as context for a specific application in advance,
- context is stable, i.e. while context may vary from application to application, it does not vary from instance to instance of an interaction with an application;
- context and activity is separable, i.e. context is taken to describe features of the environment within which an activity takes place but the elements of the activity do not belong to context itself.

So far we presented a number of techniques for building good interfaces. We also saw how the view of embodied interaction can be used as a paradigm for ubiquitous computing. In general, a technical solution can be called *intelligent* for two reasons:

(i) there is some built-in intelligent computation that solves some otherwise unsolvable problem;
(ii) using the system, a user can solve an otherwise unsolvable problem, even though the system itself does not actually do anything intelligent.

Suppose that calculating the logarithm of a number is a hard problem for a human, then a calculator is a good example for case (i) and an abacus would be an example for (ii). The calculator solves the problem for the human and the abacus empowers users to solve the problem on their own. It has also been pointed out that the classical approach of artificial intelligence is a rationalist one and according to this approach a system should model the knowledge that human experts have and thus emulate human intelligence. In this sense, the *intelligence* moves from the user to the system. This approach is valid for many cases, e.g., if expert knowledge is rare and non-experts should also be able to work with a system. As discussed above, the interactional approach seeks to make the interaction more intelligent. This fits to many new trends in artificial intelligence where embodied intelligence is viewed as a property that emerges from the interaction of an intelligent agent with the environment.

In this view, even communities of simple and light-weight agents can cooperate to perform intelligent behavior and can produce emergent interaction forms without full reflective and conscious knowledge of the world [29, 30]. Also from this perspective all of the above-mentioned material already describes how to build an intelligent ubiquitous interface. Because the processes for designing human-centered systems are just the right techniques for designing intelligent interactive systems, we already defined to an extend how to build intelligent user interfaces. Instead of leaving all intelligence to either the system, the user or the interaction, we can also try to get the best of all worlds and combine these techniques to a cooperative system where both the system and the user cooperate with their knowledge on solving some tasks supported by intelligent interaction techniques.

4 Concluding Remarks

In the sense discussed above, we can make the system more intelligent by enabling the system, the interaction and the user to find appropriate forms, mappings and functions in a cooperative context-adaptive manner. Intelligent user interfaces techniques exist for implementing such a blueprint adhering to the outlined principles. More principled techniques can also now be used to put more knowledge and reasoning into the system. Besides state of the art methods such as data bases, expert systems, heuristic search and planning, a number of more recent developments have caught a lot of interest by researchers and practitioners in the field. Especially the development of foundational infrastructures [17], dedicated and re-usable design patters [10] for modeling world knowledge in formal ontologies make it feasible to describe not only ground domain models, but also contextually reified descriptive models without resorting to higher logics [11]. Semantic technologies and formal models of world knowledge, therefore, had a great renaissance in the last couple of years. In context of the Semantic Web project [3], ontologies have been established as a standard method for capturing complex relations of objects and events in the world. Ontologies have also been successfully used in user interfaces in order to give the system a better understanding on the domain of an interaction and for understanding contextual information supplied by web services and *wrapped* sensors [21].

In terms of implementing a more interactional context-adaptivity that includes user and situation as well as prior interactions in addition to ground and pragmatic models of the domain at hand [24] several viable and more dynamic representational approaches have been forthcoming [23]. Hereby, among the costs of enabling interactional context-adaptivity is that the formal notion of correctness is replaced by one of *plausibility*. As we have discussed context plays a central in ubiquitous computing. Context-dependent user interfaces can greatly enhance the usability of these systems. However, context can also be challenging because it can depend on a huge number of parameters and it is still hard to acquire the needed formalization of the meaning of contexts and to learn the relations between descriptive and ground entities autonomously. To remedy this new approaches, e.g. in the form of human computation [2], can be applied [31], as well as so-called category games [26].

References

1. ADELSTEIN, F., GUPTA, S., RICHARD, G., AND SCHWIEBERT, L. *Fundamentals of Mobile and Pervasive Computing*. McGraw Hill, 2004.
2. AHN, L. V. Human computation. In *K-CAP '07: Proceedings of the 4th international conference on Knowledge capture* (New York, NY, USA, 2007), ACM, pp. 5–6.
3. BERNERS-LEE, T., HENDLER, J., AND LASSILA, O. The semantic web. *Scientific American May* (2001).
4. CARROLL, J. *Human-Computer Interaction in the New Millennium (ACM Press)*. Addison-Wesley Professional, 2001.
5. CLARK, A. An embodied cognitive science? *TRENDS in Cognitive Sciences 3*, 9 (1999), 345–351.
6. DIX, A. J., FINLAY, J., AND ABOWD, G. D. *Human-computer interaction*, 3. ed. ed. Pearson Prentice-Hall, Harlow, 2004.
7. DOURISH, P. *Where the Action Is: The Foundations of Embodied Interaction*. The MIT Press, October 2001.
8. DOURISH, P. What We Talk About When We Talk About Context. *Personal and Ubiquitous Computing 8*, 1 (Feb. 2004), 19–30.
9. DOURISH, P. Responsibilities and implications: Further thoughts on ethnography and design. In *Proc. ACM Conf. Designing for the User Experience DUX 2007* (Chicago, IL, 2007).
10. GANGEMI, A. Ontology design patterns for semantic web content. In *M. Musen et al. (eds.): Proceedings of the Fourth International Semantic Web Conference* (2005), Berlin, Springer.
11. GANGEMI, A., AND MIKA, P. Understanding the semantic web through descriptions and situations. In *Proceedings of the ODBASE Conference* (2003), Springer.
12. GIBBS, R. *Embodiment and Cognitive Science*. Cambridge University Press, 2005.

13. HEIDEGGER, M. *Einfuhrung in die Metaphysik. English.* Yale University Press, 1959.
14. LAKOFF, G., AND JOHNSON, M. *Metaphors We Live By.* University of Chicago Press, 1980.
15. MALAKA, R. Intelligent user interfaces for ubiquitous computing. In *Handbook of Research: Ubiquitous Computing Technology for Real Time Enterprises*, D. I. G. Max Mhlhuser, Ed. Springer, Heidelberg, Germany, 2008.
16. MALAKA, R., HAEUSSLER, J., ARAS, H., MERDES, M., PFISTERER, D., JOEST, M., AND PORZEL, R. Intelligent interaction with a mobile system. In *SmartKom - Foundations of Multimodal Dialogue Systems*, W. Wahlster, Ed. Springer, Berlin, 2006, pp. 505–522.
17. MASOLO, C., BORGO, S., GANGEMI, A., GUARINO, N., OLTRAMARI, A., AND SCHNEIDER, L. Wonderweb deliverable d17: The wonderweb library of foundational ontologies, 2003. http://wonderweb.semanticweb.org/deliverables/documents /D18.pdf (last accessed: 04/18/2009).
18. MUDERSBACH, G., KHAN, A. M., SHARMA, G., SAYAH, S. A., AWAD, M., AND OTHMAN, A. Intelligent interaction. In *Proceedings of the fifth Student Interaction Design Research Conference* (Eindhoven, the Netherlands, April 2009), pp. 22–23.
19. NORMAN, D. *Psychology of Everyday Things.* Basic Books, New York, 1988.
20. NORMAN, D. Affordances, conventions, and design. *Interactions 6*, 3 (1999), 38–41.
21. OBERLE, D., LAMPARTER, S., EBERHART, A., AND STAAB, S. Semantic management of web services. In *Proc. of ICSOC-2005 - 3rd Int. Conference on Service Oriented Computing, Amsterdam, The Netherlands* (12 2005).
22. PORZEL, R., AND GUREVYCH, I. Towards context-adaptive utterance interpretation. In *Proceedings of the 3rd SIGdial Workshop,* Philadelphia, USA, 2002 (2002), pp. 90–95.
23. PORZEL, R., GUREVYCH, I., AND MALAKA, R. In context: Integrating domain- and situation-specific knowledge. In *SmartKom - Foundations of Multimodal Dialogue Systems*, W. Wahlster, Ed. Springer Verlag, 2006, pp. 269–284.
24. PORZEL, R., ZORN, H.-P., LOOS, B., AND MALAKA, R. Towards a seperation of pragmatic knowledge and contextual information. In *ECAI'06 Workshop on Context and Ontology, Riva del Garda, August 28,* (2006).
25. PREECE, J., ROGERS, Y., AND SHARP, H. *Interaction Design : beyond human - computer interaction.* Wiley, New York, NY, 2002.
26. PUGLISI, A., BARONCHELLI, A., AND LORETO, V. Cultural route to the emergence of linguistic categories. *Proceedings of the National Academy of Sciences* (June 2008), 0802485105+.
27. SEBANZ, N., BEKKERING, H., AND KNOBLICH, G. Joint action: Bodies and minds moving together. *Trends in Cognitive Sciences 10* (2006), 70–76.
28. SHNEIDERMAN, B. *Designing the User Interface.* Addison Wesley, 1997.
29. STEELS, L. The origins of ontologies and communication conventions in multi-agent systems. *Autonomous Agents and Multi-Agent Systems 1* (1998), 169–194.
30. STEELS, L. Spontaneous lexicon change. In *In Proceedings of COLING-ACL* (1998), ACL, pp. 1243–1249.
31. TAKHTAMYSHEVA, A., PORZEL, R., AND KRAUSE, M. Games for games: Manipulating context in human computation games. In *KDD'09 Workshop on Human Computation, Paris, June 28,* (2009).
32. WAHLSTER, W. Smartkom: Symmetric multimodality in an adaptive and reusable dialog shell. In *Proceedings of the of the 26th German Conference on Artificial Intelligence, September 2003, Hamburg, Germany* (2003), DLR, Ed.
33. WEISER, M. The computer for the 21st century. *Scientific American 265*, 3 (September 1991), 66–75.
34. WEISER, M. Creating the invisible interface. In *Proceedings of the 7th annual ACM symposium on User interface software and technology* (1994), P. Szekely, Ed., ACM Press.