A Survey of Proactive Pervasive Computing

Sebastian VanSyckel
University of Mannheim
Schloss
68161 Mannheim, Germany
sebastian.vansyckel@uni-mannheim.de

Christian Becker
University of Mannheim
Schloss
68161 Mannheim, Germany
christian.becker@uni-mannheim.de

UbiComp'14 Adjunct, September 13 - 17, 2014, Seattle, WA, USA

ACM 978-1-4503-3047-3/14/09...$15.00.
http://dx.doi.org/10.1145/2638728.2641672

Abstract
Pervasive computing applications are context-aware and adapt in order to cope with changes in their environment. In this, they should be as unobtrusive as possible. Proactive computing aims at acting on behalf of the user. Proactive adaptation allows to change the application and/ or the context based on prediction. In this paper, we discuss and classify proactive pervasive computing research, as well as give an outlook on the field.

Author Keywords
Proactive Adaptation, Context-aware Systems and Applications, Pervasive Computing

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous.

Introduction
Adaptation is key to conquer the dynamism and complexity of context-aware systems, e.g., in pervasive and ubiquitous computing. Most of adaptive systems have been designed to be reactive, i.e., they react to changes in the context. However, if the reason for adaptation can be predicted, new options become available. A system can precalculate configurations in order to reduce the delay due to a necessary adaptation, or even avoid a specific

adaptation entirely. That is, by eliminating the reason for an adaptation – e.g., adjusting the context at a location before the user arrives, or even rerouting the user around an unadaptable context – we can make observable adaptations completely disappear.

Extensive surveys of the field of context-aware computing, context prediction, and adaptive architectures can, for example, be found in [2], [5], [7], [10], [15], [18], and [28]. In light of the great potential and continuous advances, there is equal cause for a survey of proactive context-awareness and adaptation. This paper presents a classification of proactive pervasive computing research and shows roadmaps for further research. We discuss the relation between proactive computing and proactive adaptation, derive a categorization of application-level proactive adaptation, give an overview of research in the field, and classify the projects following our categorization.

The remainder of the paper is structured as follows. First, we give a short background on proactive computing and compare it with proactive adaptation. Second, we discuss research from proactive context-awareness, as well as proactive adaptation, before classifying the works. Finally, we close with an outlook on the field.

Proactive Computing

The term *proactive computing* was first introduced by Tennenhouse [30] in 2000. It describes the evolution away from *interactive computing*, i.e., from classical human-centered workstation settings to human-(un)supervised pervasive computing scenarios. In [35], Want *et al.* further discuss proactive computing as well as the differences to *autonomic computing*. The aim of proactive computing are unobtrusive systems that connect to the physical world and require as little human

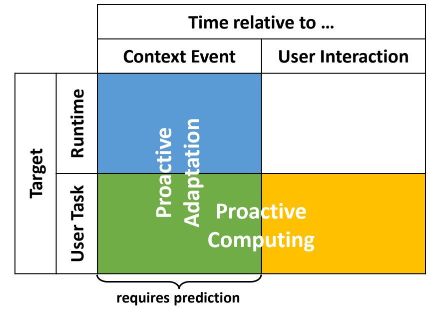

Figure 1: Relationship between Proactive Computing and Proactive Adaptation

interaction as possible. Further, they should anticipate the user's needs and act on his/ her behalf. The authors identify seven principles as foundations of proactive systems: connecting with the physical world, deep networking, macro-processing, dealing with uncertainty, anticipation, closing the control loop, and making systems personal. Despite leading to similar techniques, autonomic computing, in contrast, describes the discipline of managing the complexity of a heterogeneous system through appropriate system design principles.

In [26], Salovaara *et al.* discuss the general concept of proactive computing. They suggest that a system can act proactively, if it has a hypothesis about what its user's goals are. In order to achieve these goals, the system makes use of different resources. The authors present a classification of six different types of proactive resource management in order to become a proactive system: preparation, optimization, advising, manipulation, inhibition, and finalization of user's resources.

Handte *et al.* describe proactivity from an adaptation perspective as *modifications of an application performed* *before an application can no longer be executed* [15]. In our PROACTIVE project [33], we further explicitly include context adaptation as a necessity, in order to be able to avoid having to adapt the application itself. This view is not necessarily congruent with Tennenhouses proactive computing. As an example, in [31], the system automatically adjusts the lighting of the environment based on what it anticipates the users desire is. Hence, it acts on the users behalf. However, it does so after it notices a change, i.e., in a reactive manner from an adaptation standpoint. Hence, proactive can refer to *before the user acts*, or *before the triggering event happens*, respectively. The main difference is that in order to act before an event takes place, the system must have knowledge of that event and, hence, requires prediction. Figure 1 depicts this relationship between proactive computing and proactive adaptation.

Proactive Context-aware Applications

A comprehensive survey of the field of context-aware computing can, for example, be found in [18]. From the literature, there are four types of context-aware applications [12, 4], namely

- **Presentation** of information and services to a user,
- **Selection** of information and services,
- **Tagging** of context to information for later retrieval, and
- **Automatic execution** of a service for a user,

that are on a range between *active* and *passive context-awareness* [10].

- **Passive context-awareness**: An application presents the new or updated context to an interested user or makes the context persistent for the user to retrieve later.
- **Active context-awareness**: An application automatically adapts to discovered context, by changing the applications behavior.

In recent years, there has been an increasing number of works with proactive characteristics. The majority is in the field of so called proactive services, e.g., [17, 23, 9, 22]. Here, services are selected based on the current context of the user as well as his/ her preferences. The user preferences are mostly not explicitly specified by the users themselves, but rather reasoned about based on previous user behavior, which achieves a higher level of unobtrusiveness. Subsequently, the list of selected services are proactively suggested to the user, i.e., without the user having to request them. Next we discuss research on proactive context-aware applications in pervasive systems.

In Proactive Sensor Networks (PSNs) [1], sensors and actuators form so called context overlays in order to make distributed decisions, i.e., without a centralized entity, such as a context management, in the system. The network is proactive in that the sensors themselves – by themselves or in collaboration – preprocess their sensed data to the needs of the respective actuators. Such collaboration of sensors, e.g., in order to infer high from low level context, is referred to as *distributed context decisions*. Not being a traditional context-aware application, the approach best fits in the category of context-aware presentation, even though the users in this work are actuators rather than humans.

The first work on proactive service selection was presented by Pitkäranta *et al.* in 2005 in the DYNAMOS project [23]. The system matches services based on their description to the context of its users, and proactively notifies them about critical situations of predefined interests. In [25], the authors report on their experiences of applying DYNAMOS for the boaters in a marine community. Examples for selected information are discounts on gasoline or location-aware wind warnings.

In [22], Pawar *et al.* present a geographically distributed patient monitoring system that utilizes proactive service selection techniques for their context-aware emergency response service (ERS). The approach is limited to the ERS, which makes it application-specific.

In [17], Hong *et al.* present an agent-based framework, that aims at providing personalized services to its users based on their respective context history. It is proactive in the sense that it does this automatically using context reasoning, instead of being based on manually defined preferences. Therefore, the main challenge in this work is to infer the user preferences for the current situation from past situations, and then find a matching service.

Similarly, ProWMS [9] is a context-aware web service pre-fetching strategy for mobile devices based on user preferences. The preferences are automatically computed from the service request history considering the current context.

In [24], Rasch *et al.* present an approach to proactive service selection specifically for pervasive environments. Central to the work is their formal context model called *Hyperspace Analogue to Context* (HAC). They model the capabilities of services as well as user preferences as multidimensional geometric structures. The relation between these structures is used to calculate a service to preference matching metric. Finally, the system presents a ranked list of services to the user.

Another example for proactive service discovery is the work by Fenza *et al.* [13], which focuses on the healthcare domain. It aims at providing personalized services depending on the patient's state of health, which is acquired from a network of wearable sensors. The two main technologies are Semantic Web for service and context modeling – specifically OWL-S – and fuzzy logic for context reasoning. Again, the system's output is a ranked list of services that match the patient's health context.

In the MavHome project [11], Cook *et al.* explore an agent-based smart home management system that acts on the user's behalf or supports the user in his/ her task. The agents have a set of defined actions that are triggered based on the predicted next user task/ goal, such as, for example, starting the lawn sprinklers or placing a food order. For this, it combines the strength of several prediction approaches into the meta predictor called Predict^2. MavHome offers implicit context adaptation via actuators, such as automated blinds, but is an action-based framework and not an adaptive architecture. That is, the system does not feature a control loop that specifically monitors and adapts a certain context, but aims at automating the actions that the inhabitant would instead have to carry out.

Proactive Adaptation

The capability of a mobile application to dynamically adapt to its runtime environment allows it to use resources beyond the boundary of the device it is running on. Instead, it can use resources and services in its environment, such as, for example, speech input, video output, or a temperature regulating service. The adaptation *control* in such systems ranges from manual adaptation by the user to automatic adaptation by the underlying system. Satyanarayanan first refers to this as the application-awareness of adaptation [27], with the *laissez-faire* strategy representing adaptation on the application side, and the *application-transparent* strategy representing adaptation by the system.

However, further research in the field has shown that a more specific classification of laissez-faire adaptation is needed. Manual adaptation requires interaction from the user, whereas adaptations that an application automatically instructs – e.g., based on user preferences – are human supervised. In [3], Becker *et al.* address the need and describe three levels of adaptation support:

1. **Manual adaptation:** The application presents adaptation possibilities to the user, who makes the decision.
2. **Application-specific automatic adaptation:** Each application has its own automated adaptation routines, which were implemented by the programmer. Although this approach increases user experience, predefined and, therefore, inflexible routines bear issues in changing environments.
3. **Generic automatic adaptation:** Application programmers define a set of required services their application needs for execution. The service provision, however, is in the responsibility of the underlying system, e.g., the operating system (OS) or middleware. Additionally, the user may be able to list preferences, which influence the decision in case of multiple options.

Next to control, adaptation in pervasive systems is further characterized by the dimensions *technique*, *level*, and *time* [15]. Adaptation technique distinguishes between the adaptation of a component's behavior, its composition, and the adaptation of the context itself. Adaptation level, on the other hand, differentiates adaptation at the network, system, and application level. Typically, network and system-level adaptations are reactions to broken links, high latencies, etc., and are not directly related to the user. Hence, we focus on application-level adaptation. Finally, time separates proactive from reactive adaptation. Figure 2 shows the resulting categories of application-level proactive adaptation.

Application-level Proactive Adaptation	Technique	Behavior
		Composition
		Context
	Control	Manual
		Application-specific automatic
		Generic automatic

Figure 2: Categories of Application-level Proactive Adaptation

Next, we discuss research on proactive adaptation in pervasive systems. We include the *act on behalf*-view of proactivity and indicate, which projects use predictions in order to adapt before the event.

In his dissertation [20], Mayrhofer presents a general architecture for context prediction that allows applications to query future context information in order to proactively assist its users. The focus of the work is a five step prediction process: (1) gather context information from a heterogeneous network of sensors, (2) extract features, (3) classify, (4) label, and (5) predict. However, the framework specifically offers an interface for applications to access predictions and, thus, is a context-aware application/ adaptation framework.

Prism [29] – the task manager in the Aura [14] architecture – prepares adaptations on the application level that are related to the next anticipated user tasks.

For example, the system uses approximate location predictions based on location data in calendar entries in order to prepare data transfer to a specific location. For this, Prism uses a set of tasks to application mappings, e.g., *edit text* to Microsoft Word, that are provided by the system administrator. The adaptation itself is then triggered by the user.

iROS [19] supports proactive adaptation of the composition of its loosely coupled applications. That is, iROS' event heap selects appropriate components from its repository based on their descriptions. In case of errors, for example crash of a component, the system resends events to an alternative component with the same capabilities, if available.

Adaptable Pervasive Flows (APFs) [16] are workflow-like models of an entity's activity that adapt with regard to that entity's situation. The flows consist of a series of tasks – either representing atomic services or a *subflow* – that are connected by so-called *context-aware transitions*, i.e., transitions defined by context events, such as a location change. Adaptation in the APF system is generic-automatic. Developers define a set of goals and constraints for each flow, and the *flow system* calculates the specific adaptations of a flow's composition. These are either *horizontal* adaptations of the flow in case re-planning is necessary, or *vertical* adaptations in the case that a task is substituted by a subflow or it's mapping to an atomic service became invalid. The APF system uses context prediction in order to anticipate these adaptations. APFs are part of the EU-funded project ALLOW.

In [8], Byun *et al.* present an approach that learns user preferences for human supervised environment control from context history. The system is application-specific automatic in that it is not a framework for applications, but monitoring and adapting the context is the application itself. The authors further discuss different uncertainty issues in general with regard to control and adaptation, and propose possible solutions for these problems.

Similarly, Vainio *et al.* [31] present proactive context adaptation based on fuzzy logic techniques for smart home environments. That is, the system monitors the state of an adaptable context, such as lighting and temperature, learns the user's routines, and subsequently adapts the context under the user's supervision. Due to the use of fuzzy rules, the system can control the environment even in situations of uncertainty, i.e., ones it has not learned yet. As in the work by Byun *et al.*, controlling the environment is the application itself.

In [6], Boytsov and Zaslavsky present the CALCHAS system, which offers context predictions to applications that, in turn, use actuators to adapt their context. In order to support proactive adaptation, they further present an extension to the context spaces theory [21] with the concept of context adaptation via actuators. The authors see proactive adaptation as reinforcement learning tasks that aim at improving both the predictions as well as the adaptation decisions. Actually, the focus of the work is on the quality of the predictions and decisions, and not on the framework for adaptation. The system follows the laissez-faire/ application-specific automatic approach to application adaptation, as it provides the predicted context information to the applications, but does not adapt the context itself.

COMITY [32] – part of the 3PC project [15] – is a framework for adaptation coordination in pervasive systems. Based on a set of possible runtime configurations that are reported by the applications in the environment, the system finds optimal interference-free global

Project	**Service**			**Technique**			**Control**			**Uses Context Prediction?**
	pre	sel	exe	beh	com	ctx	man	app	gen	
PSNs	+							+		
DYNAMOS		+						+		
Pawar *et al.*		+						+		
Hong *et al.*		+							+	
ProWMS		+							+	
Rasch *et al.*		+							+	
Fenza *et al.*		+							+	
MavHome		(+)	+					+		✓
Mayrhofer				+				+		✓
Prism (Aura)					+		+			✓
iROS					+				+	
APFs (ALLOW)					+				+	✓
Byun *et al.*						+		+		
Vainio *et al.*						+		+		
CALCHAS						+		+		✓
COMITY (3PC)				(+)	(+)	+		+	(+)	
PROACTIVE (3PC)				*	+/*	+		*	+	✓

+,* = Feature and respective control
(+) = Implicit feature

Table 1: Classification of Proactive Pervasive Computing Research

configurations and instructs necessary adaptations. This helps avoiding oscillating adaptations, without the users having to adapt their applications manually. Even though the system calculates the *adaptation plans* – instructions to switch into one of the specified configurations – the respective applications have to execute the adaptations themselves. The aim of these adaptations are the effects of an application on the shared context. However, the behavior and composition may have to change as well.

In our PROACTIVE project, we are developing a framework for generic automatic proactive adaptation in pervasive systems. In order to translocate adaptation into the system, we developed a variable-based context management and interaction entity [34]. Using an application's requirements specification, the *configuration management* pre-calculates and rates all possible adaptation alternatives for the application with regard to its predicted context [33]. As soon as the predicted context event occurs, the system triggers the prepared application adaptation (behavior and composition) and adapts the application's context.

Classification

Table 1 shows a classification of proactive pervasive computing research with regards to their application (*presentation*, *selection*, and *automatic execution*), or adaptation technique (*behavior*, *composition*, and *context*), respectively, as well as control (*manual*, *application-specific automatic*, and *generic automatic*) and the use of context prediction.

As mentioned before, most proactive context-aware application systems are in the category of service selection. They use learning-based algorithms to reason about the users preferences, use this information to filter out unwanted services, and let the user choose which service should be used. In the field of proactive adaptation, application-specific automatic adaptation of an application's structure and its context are prevalent.

Outlook

Generic automatic proactive adaptation, i.e, in the responsibility of the underlying system software, has not yet been extensively researched. However, as shown in the domain of reactive adaptive architectures, generic automatic approaches are the foundation of robust pervasive environments. Such systems and frameworks allow application developers to focus on the application logic instead of dealing with the uncertainty of the application's runtime environment. Especially suitable architectures, interfaces and adaptation strategies that support proactive adaptation of an application's behavior and/ or composition should be one of the focuses of research in the field of adaptive architectures.

For proactive systems that aim at adapting applications and/ or context before it becomes necessary, high accuracy context prediction – i.e., reliable probabilities and handling of false positives/ negatives – is key. Much research has been done in the field, leading to very good predictors regarding location, the next user task, or sequence of upcoming events. Comprehensive surveys on state-of-the-art prediction methods can, for example, be found in [7] and [28]. However, dead reckoning approaches aside, there remains a great need for point-in-time prediction, i.e., *when* an event will happen, with an accuracy-range of a few seconds. This fact leads to proactive systems in practice preparing for context events and then triggering the adaptation as soon as the event happens, instead of truly adapting ahead of time. If achieved, such predictors would greatly advance the field.

Acknowledgements

This work is supported by the German Research Foundation (DFG).

References

[1] Ahn, S., and Kim, D. Proactive context-aware sensor networks. In *Wireless Sensor Networks*. Springer, 2006.

[2] Baldauf, M., Dustdar, S., and Rosenberg, F. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing 2*, 4 (2007).

[3] Becker, C., Handte, M., Schiele, G., and Rothermel, K. Pcom - a component system for pervasive computing. In *Proc. Pervasive Computing and Communications (PerCom)*, IEEE (2004).

[4] Becker, C., and Nicklas, D. Where do spatial context-models end and where do ontologies start? a proposal of a combined approach. In *Proc. Workshop on Advanced Context Modelling, Reasoning and Management* (2004).

[5] Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., and Riboni, D. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing 6*, 2 (2010).

[6] Boytsov, A., and Zaslavsky, A. Extending context spaces theory by proactive adaptation. In *Smart Spaces and Next Generation Wired/Wireless Networking*. Springer, 2010.

[7] Boytsov, A., and Zaslavsky, A. Context prediction in pervasive computing systems: Achievements and challenges. In *Supporting Real Time Decision-Making*. Springer, 2011.

[8] Byun, H. E., and Cheverst, K. Supporting proactive intelligent behaviour: the problem of uncertainty. In *Proc. Workshop on User Modelling for Ubiquitous Computing* (2003).

[9] Chang, C., Ling, S., and Krishnaswamy, S. Promws: Proactive mobile web service provision using context-awareness. In *Proc. Pervasive Computing and Communications Workshops*, IEEE (2011).

[10] Chen, G., and Kotz, D. A survey of context-aware mobile computing research. Tech. rep., Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.

[11] Cook, D. J., Youngblood, M., Heierman III, E. O., Gopalratnam, K., Rao, S., Litvin, A., and Khawaja, F. Mavhome: An agent-based smart home. In *Proc. Pervasive Computing and Communications (PerCom)*, IEEE (2003).

[12] Dey, A. K., and Abowd, G. D. Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing*, Springer (1999), 304–307.

[13] Fenza, G., Furno, D., and Loia, V. Enhanced healthcare environment by means of proactive context aware service discovery. In *Proc. Advanced Information Networking and Applications (AINA)*, IEEE (2011).

[14] Garlan, D., Siewiorek, D. P., Smailagic, A., and Steenkiste, P. Project aura: Toward distraction-free pervasive computing. *IEEE Pervasive Computing 1*, 2 (2002).

[15] Handte, M., Schiele, G., Matjuntke, V., Becker, C., and Marrón, P. J. 3pc: System support for adaptive peer-to-peer pervasive computing. *ACM Transactions on Autonomous and Adaptive Systems 7*, 1 (2012).

[16] Herrmann, K., Rothermel, K., Kortuem, G., and Dulay, N. Adaptable pervasive flows-an emerging technology for pervasive adaptation. In *Proc. Self-Adaptive and Self-Organizing Systems Workshops (SASOW)*, IEEE (2008).

[17] Hong, J., Suh, E., Kim, J., and Kim, S. Context-aware system for proactive personalized service based on context history. *Expert Systems with Applications 36*, 4 (2009).

[18] Hong, J., Suh, E., and Kim, S. Context-aware systems: A literature review and classification. *Expert Systems with Applications 36*, 4 (2009).

[19] Johanson, B., Fox, A., and Winograd, T. The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE Pervasive Computing 1*, 2 (2002).

[20] Mayrhofer, R. *An Architecture for Context Prediction*. PhD thesis, Johannes Kepler University Linz, 2004.

[21] Padovitz, A., Loke, S. W., and Zaslavsky, A. Towards a theory of context spaces. In *Proc. Pervasive Computing and Communications Workshops*, IEEE (2004).

[22] Pawar, P., van Beijnum, B.-J., Mei, H., and Hermens, H. Towards proactive context-aware service selection in the geographically distributed remote patient monitoring system. In *Proc. Wireless Pervasive Computing (ISWPC)*, IEEE (2009).

[23] Pitkäranta, T., Riva, O., and Toivonen, S. Designing and implementing a system for the provision of proactive context-aware services. In *Proc. Workshop on Context Awareness for Proactive Systems (CAPS)* (2005).

[24] Rasch, K., Li, F., Sehic, S., Ayani, R., and Dustdar, S. Context-driven personalized service discovery in pervasive environments. *World Wide Web 14*, 4 (2011).

[25] Riva, O., Pitkäranta, T., and Toivonen, S. Proactive context-aware service provisioning for a marine community. 175.

[26] Salovaara, A., and Oulasvirta, A. Six modes of proactive resource management: A user-centric typology for proactive behaviors. In *Proc. Nordic Conference on Human-computer Interaction*, ACM (2004).

[27] Satyanarayanan, M. Mobile computing: Where's the tofu? *ACM SIGMOBILE Mobile Computing and Communications Review 1*, 1 (1997).

[28] Sigg, S. *Development of a novel context prediction algorithm and analysis of context prediction schemes*. PhD thesis, Kassel University, 2008.

[29] Sousa, J. P., and Garlan, D. Aura: An architectural framework for user mobility in ubiquitous computing environments. In *Software Architecture*. Springer, 2002.

[30] Tennenhouse, D. Proactive computing. *Communications of the ACM 43*, 5 (2000).

[31] Vainio, A.-M., Valtonen, M., and Vanhala, J. Proactive fuzzy control and adaptation methods for smart homes. *IEEE Intelligent Systems 23*, 2 (2008).

[32] VanSyckel, S., Schäfer, D., Majuntke, V., Krupitzer, C., Schiele, G., and Becker, C. Comity: A framework for adaptation coordination in multi-platform pervasive systems. *Pervasive and Mobile Computing 10* (2014).

[33] VanSyckel, S., Schafer, D., Schiele, G., and Becker, C. Configuration management for proactive adaptation in pervasive environments. In *Proc. Self-Adaptive and Self-Organizing Systems (SASO)*, IEEE (2013).

[34] VanSyckel, S., Schiele, G., and Becker, C. Extending context management for proactive adaptation in pervasive environments. In *Ubiquitous Information Technologies and Applications*. Springer, 2013.

[35] Want, R., Pering, T., and Tennenhouse, D. Comparing autonomic and proactive computing. *IBM Systems Journal 42*, 1 (2003).