

Core Services for Smart Products

Markus Miche¹, Daniel Schreiber², and Melanie Hartmann²

¹ SAP Research CEC Zurich, Kreuzplatz 20, 8008 Zurich, Switzerland,
markus.miche@sap.com

² Telecooperation Group, Technische Universität Darmstadt,
Hochschulstr. 10, 64289 Darmstadt, Germany
{schreiber, melanie}@tk.informatik.tu-darmstadt.de

Abstract. The increasing complexity and diversity of technical products plus the massive amount of product-related data overwhelms humans dealing with them at all stages of the life-cycle. We present a novel architecture for building *smart* products that are able to interact with humans in a natural and proactive way, and assist and guide them in performing their tasks. Further, we show how communication capabilities of smart products are used to account for the limited resources of individual products by leveraging resources provided by the environment or other smart products for storage and natural interaction.

1 Introduction

The complexity (i.e., functions and features) and proliferation of customization options of technical products starting from consumer goods such as cell phones all the way up to cars and airplanes is constantly increasing; the amount of product-related data that needs to be managed is growing massively. Due to this trend, users are overwhelmed at all phases of the product life-cycle, especially during manufacturing, maintenance, and use. For brevity, we refer to humans interacting with products as *users*, regardless of the life-cycle step this interaction takes place in. Adopting ubiquitous computing principles and making products *smart* can solve these problems. Smart products assist their users during the whole life-cycle, literally talking to and guiding them to deal with their complexity. Two key challenges to realize the vision of smart products are (i) to support natural interaction with the user and (ii) to make use of other smart products and resources available in the environment. The latter is essential as smart products are usually limited in their integrated interaction capabilities, context sensing functionality, and storage capacities.

Existing architectures for smart products focus on making individual products smart [1–3] without addressing the need for communication with other products and interacting with users in the environment. This distinguishes our proposed architecture for smart products from architectures for sensor networks, which do not support direct interaction with human users. Following the ideas of [4], the integration of smart products into business processes is treated in [5]. However, the interaction with the user is very restricted as products are merely

able to signal warnings on the user’s cell phone. The architecture presented in [6] centers on natural interaction between products in a shop and users using NLP techniques. This approach is not applicable to many smart products environments. In cases where the tasks desired by the user can be performed by other smart products, interaction with the user should be avoided and the necessary actions should be automatically invoked on other smart products.

We present a novel architecture for smart products that addresses the challenges described above. Thereby, three core services for smart products and their architectural relations are pointed out: Context sensing, user interaction, and distributed storage of data.

2 Architecture

A high level view of the proposed architecture for smart products is depicted in Figure 1 (undirected links represent a bi-directional information flow). Every smart product can comprise input and output capabilities, actuators to trigger its built-in functionality, sensors, as well as product-specific data. To connect to resources in the environment, a communication module is integrated into each smart product. The communication module is based on the middleware MundoCore [7], which allows products to discover and communicate with each other in a peer-to-peer way. In the following, we give a short description on how other components utilize this communication module to realize the challenges (i) and (ii).

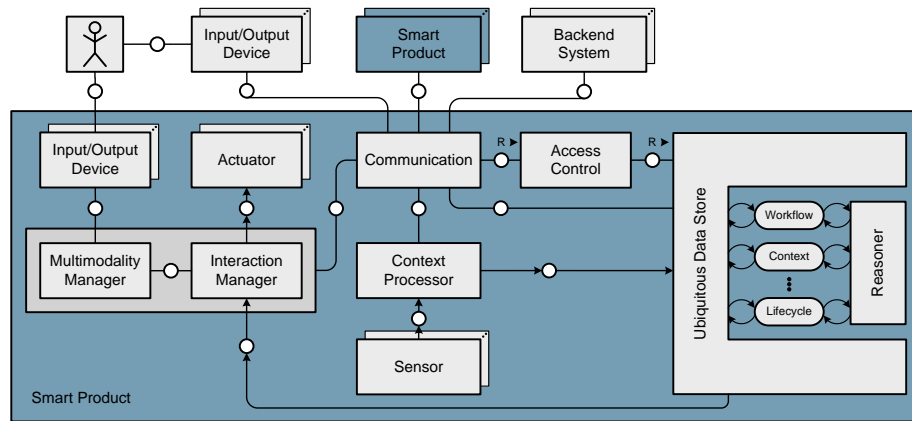


Fig. 1: Simplified Overview of the Proposed Architecture for Smart Products

Context To interact naturally with the user, products must be aware of their current *context*. We consider two facets of context awareness: Acquiring context and reacting to context, one without the other is meaningless. For that purpose, each smart product needs to define its own rules on how to react in a given context. An example of a context aware product is a coffee machine that turns off

the heating plate, if a temperature above a critical threshold is reported by an internal temperature sensor. However, this context is very basic. Much smarter behavior can be triggered if *higher level context* is inferred from this low level context, e.g., starting to prepare coffee if the coffee machine knows that its owner will soon have breakfast. In almost all cases such higher level context needs input from a variety of different sensors, physical as well as virtual ones. Since equipping each smart product with all the necessary sensors is infeasible, it is important that smart products can gather context information in a distributed way. Therefore, the *Context Processor* component on each smart product is not only connected to local sensors but can further subscribe to context information provided by other smart products in the environment using the communication middleware. As a result, the distributed context is richer than the context acquired by a single smart product using locally connected sensors only.

Interaction The main goal of making products smart is to facilitate interaction for the user as much as possible. This comprises (i) automating workflows in order to avoid interaction, (ii) proactively guiding the user through non-automatable workflows, and (iii) providing natural interaction in case no workflow is followed by the user. For automatically executing (parts of) a workflow, smart products need to be able to discover other products in the environment that are capable of performing the respective steps of the workflow. The user should only be approached if no suitable product is available or if the user's explicit confirmation is required. However, the ability of a smart product to interact naturally is impaired by the limited input and output capabilities of typical smart products. To overcome these limitations, smart products should be able to make use of the interaction capabilities of the environment.

These requirements are reflected in our architecture as follows: The *Interaction Manager* is responsible for managing workflows to be executed. It triggers the execution of all workflow steps the smart product can perform by itself. For all other steps, it searches for suitable products in the environment. If necessary, it approaches the user by sending a user interface description to the *Multimodality Manager*, which chooses the best combination of interaction devices in the environment in order to realize the most natural way of presenting this interface to the user. Furthermore, our architecture allows smart products to initiate the interaction with a user in a proactive way. This enables the product to proactively offer help by itself and to assist the user in interacting with it.

Ubiquitous Data Store During their whole life-cycle, smart products require plenty of information such as for example technical information needed for manufacturing and maintenance activities or manuals needed by the end user. However, due to their resource constraints, smart products are in general not capable of storing all information locally. Also, it would not be reasonable to store all data in a remote storage infrastructure, because of the varying communication capabilities of smart products ranging from WiFi and mobile broadband wireless access technologies to short-range technologies such as Bluetooth or IrDA.

Therefore, we propose a storage mechanism that facilitates pre-fetching of information required in different phases of the product life-cycle based on an appropriate life-cycle model. Moreover, a data distribution concept is defined, which allows for storing information in the smart product's environment. For that purpose, dedicated metrics are defined to determine suitable products for (temporary) storing data. Since smart products can be mobile, the mechanism must further enable data stored nearby a product to follow the product to finally enhance both the availability of information and the performance of the data access. We referred to this concept as the smart product's *mobile cache cloud*.

This functionality is covered by the *Ubiquitous Data Store*, which facilitates the distribution of information among smart products plus the access to data stored in backend systems. In addition, this component provides a synchronization mechanism to ensure different levels of consistency (e.g., atomic and eventual) for data replicated and stored in the mobile cache cloud of a smart product.

3 Conclusion

Our architecture including its core services addresses the identified key challenges of smart products. Thus, it allows for natural interaction with the user and to incorporate resources available in the environment both as input and output devices and as additional storage capacities thus overcoming the limited capabilities of smart products. As a next step, we intend to implement a first prototype of the architecture to analyze whether it copes with the limited resources of smart products in terms of memory and processing power.

Acknowledgments Part of this research was conducted within the *Smart-Products* project funded as part of the Seventh Framework Programme of the EU (grant number 231204).

References

1. Kawsar, F., Nakajima, T., Park, J.H., Jeong, Y.S.: A document based framework for smart object systems. *Future Generation Communication and Networking* **1** (2008) 178–183
2. Antifakos, S., Michahelles, F., Schiele, B.: Proactive instructions for furniture assembly. In: *UbiComp*. (2002) 351–360
3. Holmquist, L.E., Gellersen, H.W., Kortuem, G., Schmidt, A., Strohbach, M., Antifakos, S., Michahelles, F., Schiele, B., Beigl, M., Maze, R.: Building intelligent environments with smart-its. *IEEE Comp. Graphics and Applications* **24**(1) (2004)
4. McFarlane, D., Sarma, S.E., Chirn, J.L., Wong, C.Y., Ashton, K.: The intelligent product in manufacturing control. *Journal of EAIA* (2002) 5464
5. Dobre, D., Bajic, E.: Smart object design for active security management of hazardous products. In: *DIPSO Workshop, Innsbruck Austria* (2007) 100–106
6. Janzen, S., Maass, W.: CoRA-Interactive communication with smart products. In: *BLOCKS Workshop*. (2008)
7. Aitenbichler, E., Kangasharju, J., Mühlhäuser, M.: Mundocore: A light-weight infrastructure for pervasive computing. *Pervasive Mob. Comput.* **3**(4) (2007) 332–361