

## Middleware for Cooperating Objects

Stephan Steglich<sup>1,2</sup>, Stefan Arbanowski<sup>2</sup>

<sup>1</sup> Technical University Berlin, Franklinstrasse 28/29, 10587 Berlin, Germany

<sup>2</sup> Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany  
steglich@cs.tu-berlin.de

**Abstract:** Recent developments in mobile communication and small computing devices have had a tremendous impact on our societies. They have brought the dream of ubiquitous computing and communication closer to reality. In the near future the communication and computing devices will enable mass-market scale ubiquitous services and applications. The main challenge will be software that fulfils the needs of personalized, ambient-aware services and applications usable anywhere, anytime. A key enabler for this vision is personal networking supporting the communication in different kinds of environments: in personal domain, in ad-hoc communities, in digital home, in networks without any infrastructure as well as in networks with operator-based infrastructures.

**Keywords:** ubiquitous computing and communication, autonomic communication, ambient awareness, middleware, mobile devices

### 1. Introduction

In our vision, ubiquitous computing and communication scenarios will be facilitated in the near future through so-called *smart embedded devices*. These devices will be the core components of a personal networking system. This system is autonomously looking for suitable peripheral devices such as displays, input devices, processors, fast access memories, and access points to communication channels.

In this way, the personal networking system can dynamically build up a cooperation of devices and objects, which are most appropriate to fulfill the end-user's demand.

### 2. Smart Embedded Devices

A Smart Embedded Device probes its surroundings for other system components in order to establish an ad-hoc community (see Figure 1). It searches for different kinds of sensors in order to extract ambient information associated with the current location. Components in the surrounding environment can also provide the means, i.e. actuators, to affect the real-world properties of the surrounding.

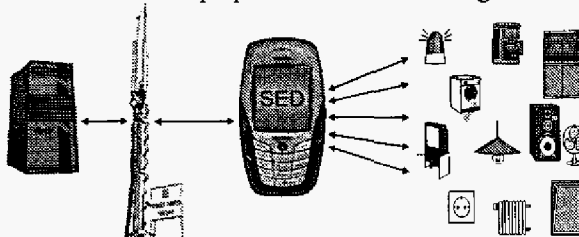


Figure 1: Smart Embedded Devices

Sensors contribute to this vision the necessary information of local context and actuators the means to affect the real-world context situation.

Such application scenarios imply a suitable service framework in which devices can provide mandatory functions and common interfaces. The current trend in developing communication software is to utilize Internet protocols; not only IP, but also Internet solutions both above and below the IP protocol. Another significant trend is the requirement of ever-faster service and application development and deployment. The immediate implication has been the introduction of various service frameworks, usually referred as middleware. The current mainstream of middleware is based on the object-oriented client/server paradigm, but is increasingly developing towards the peer-to-peer paradigm.

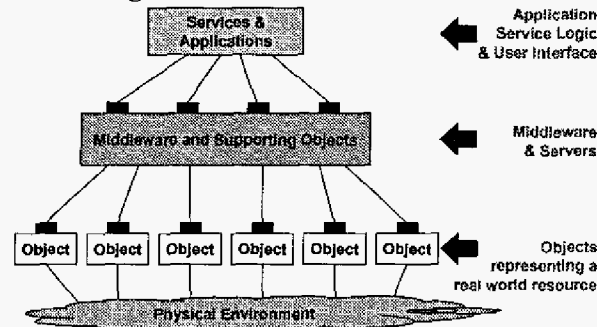
### 3. Middleware for Cooperating Objects

The goal of the Middleware for Cooperating Objects is to hide the complexity of application development for different kinds of communication environments. The focus is on ad-hoc communities (the devices, which cooperate without operator infrastructures) and in infrastructure supported networks (the devices are connected with each other and external world through communication infrastructure). Today, the solution stacks for these two environments are very different. The research challenge is to find a common light-weight solution that works reasonably well in both environments.

This middleware realizes a distributed platform of interfaces and services that reside between the applications and the devices. The devices are usually of different types and highly dynamic. This requires an according support of the middleware system, which must be able to adapt to the changing environments and to enable the inspection of the underlying components at runtime. This approach is denoted as a *reflective middleware*. The actual devices or their functional components are represented in the middleware layer as functional objects.

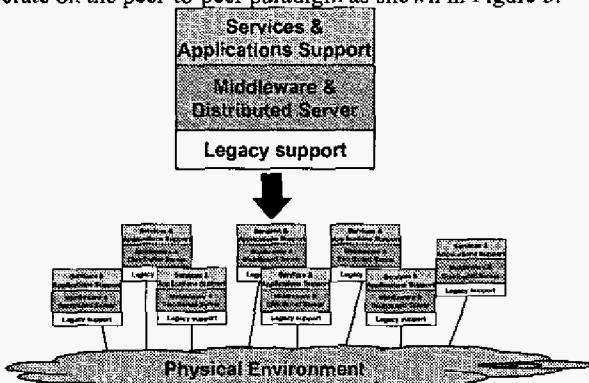
In order to realize a truly *personal* networking system, the middleware additionally needs to manage and to consider user preferences data. This means, the middleware must have a conceptual model of its current state and the preferences of the user.

Applications based on the client-server paradigm are useful and widely used but they are problematic in an ad-hoc community. In particular, most infrastructure services specified for IMT-2000 are based on the client-server paradigm following a strong hierarchical structure as shown in Figure 2.



**Figure 2: Traditional Middleware Infrastructures**

If we want smart embedded devices to work seamlessly in both environments of our focus, we need to find out how client-server applications are efficiently implemented for ad-hoc communities. When infrastructure services are not available, then the server functionality must be distributed among the members of the community in order to achieve robustness against leaving members. Here, an important challenge is the management of a distributed state. Therefore, the "server" needs to be distributed among all members of the ad-hoc community. It leads the system to operate on the peer-to-peer paradigm as shown in Figure 3.



**Figure 3: P2P community of cooperating objects**

The results of the current activities in related R&D projects, e.g. [1], are requirements, specifications, prototype implementations, and proof-of-concept demonstrations of reflective middleware solutions for smart embedded devices like PDAs and high-end mobile phones. The projects do not try to invent the wheel again, but try to reuse – probably with necessary enhancements – existing results as far as possible; particularly from the IETF (Internet protocols) and OMG (e.g. Wireless CORBA [4], Super Distributed Objects [3], Model Driven Architecture [5]).

The peer-to-peer paradigm implies a cooperative behavior of the middleware objects. A common mechanism still needs to be developed that would allow the objects to talk to each other, exchange information, and agree on how they should cooperate to accomplish global and local tasks. Additionally, a rule-based programming support is required that is capable to translate the user's wishes (global tasks) into smaller units (local sub-tasks), which can be executed by the objects. The objects also should implement a certain degree of autonomy, which can lead to a further subdivision of local tasks into even smaller tasks or to a delegation of own tasks to other objects instead of the accomplishment by itself. In this way, a hierarchical system of responsibilities and cooperations is realized, which imitates human social structures.

## 4. Summary

This paper has introduced the basic idea of a middleware for cooperating objects. According the vision of ubiquitous computing and communication scenarios based on smart embedded devices, the paper elaborates the need for a middleware for cooperating object. Functional components of the devices are represented as objects in the middleware. Due to the heterogeneity and high dynamic of these devices the middleware has to realize an adaptive behavior and to provide the inspection of the actual underlying infrastructure. This means, the envisioned middleware for cooperating objects has to implement the features of a reflective middleware.

Cooperation is to be supported by suitable mechanisms for inter-object communication and rule-based programming. In this way, middleware for cooperating objects provides a complement set of programming interfaces to the applications. Using these interfaces, smart embedded devices, such as mobile phones or PDAs, can browse their environment and search for available objects in order to delegate tasks to them. The distributed and autonomous execution of the several sub-tasks will lead to an efficient and innovative accomplishment of the user's demands.

## 5. References

- [1] Sameshima, S. et al.: *SDO and its Standardization Activity*. Proceedings of the 2004 International Symposium on Applications and the Internet Workshops (SAINTW'04), IEEE, 2004, 0-7695-2050-2/04
- [2] Object Management Group: *Super Distributed Objects*. White paper. OMG Document sdo/01-07-01.
- [3] OMG SDO DSIG. *PIM and PSM for Super Distributed Objects*. RFP. OMG Document: sdo/02-01-04
- [4] Raatikainen, K. et al.: *Wireless Access and Terminal Mobility*. White Paper, Helsinki 2001.
- [5] Miller, J.; Mukerji, J.: *Model Driven Architecture (MDA)*. Object Management Group, Juli 2001, ormsc/2001-07-01