# A novel approach to integration of mobile devices into KNX environments

Benjamin Wilms
Busch-Jaeger Elektro GmbH
Freisenbergstr. 2
D-58513 Lüdenscheid
e-mail: benjamin.wilms@de.abb.com

## Abstract

The issue of integrating mobile devices into smart environments for control purposes has been dealt with several times before. Most approaches lead to web-technology (browser visualisation) based solutions because the platform-independent thin-client concept allows the use of a broad range of devices. Other approaches are known, which have different advantages, but have to be regarded as more or less proprietary because of their platform-dependency. This novel approach aims at realising a platform-independent solution without the classical drawbacks of client-server communication in terms of real-time status consistency. For this purpose, the opportunities offered by a new API called JSR-82 are taken, i.e. using Bluetooth access from Java-Midlets in mobile devices, e.g. mobile phones.

These platform-independent applications are provided by a webservice, which takes the project information and the user's personal preferences to generate a personalised Midlet, which afterwards is sent to the user's mobile device (over-the-air provisioning, assuming a mobile phone is used). The data representation and code generation is mainly based on XML and XSL transformations. Having received the tailored application and plugged the developed module performing the Bluetooth gateway functionality onto a BCU, bidirectional real-time communication is made possible.

According to the desired seamless integration into KNX environments, the gateway module is realised as a standard KNX application module that can be plugged onto a standard BCU 1. The mobile device becomes a part of the KNX environment by being connected to this gateway module by a "Bluetooth PEI".

## Content

## 1. Motivation

Integration of mobile devices into KNX environments by wireless technologies like Bluetooth or WLAN has been dealt with several times. There are a couple of approaches to be considered, but in the end, most approaches lead to web-technology based solutions, providing the opportunity to control KNX installations by mobile devices like laptops or PDAs. One of the obvious advantages of this approach is its platform-independency, which allows the use of a broad range of devices. Laptops and PDAs are equipped with web-browsers and are integrated into KNX environments as thin clients. As a consequence, web servers have to be implemented, performing the gateway functionality to the KNX world. Although there are already very efficient solutions, energy consumption and costs are still critical issues. Furthermore, the lack of real-time status consistency is a consequence of client-server communication. The displayed content is updated upon a dedicated refresh request, but not event driven, e.g. by a manually switched light. Countermeasures like cyclic polling are not to be considered very helpful because of the increasing data traffic and energy consumption. Besides these general purpose solutions based on web-servers and client-server communication, there are also tailored solutions for dedicated devices, which obviously lead to platform-dependent realisations by definition.

With the presentation of the Java APIs for Bluetooth Wireless Technology (JSR-82, Java Service Request) a new way for integrating Java technology-enabled devices into Bluetooth environments became obvious. The use of this new API (Application Programming Interface) is especially suited for mobile phones, which are very important regarding mobile devices for control purposes in KNX environments because of their immense distribution. Java as a platform-independent technology allows the use of an increasing number of devices (e.g. BenQ P30, Motorola A1000, Nokia 6230, Nokia 6630, Nokia 7610, Nokia 9300, Sendo X, Siemens S65, Siemens SK65, Siemens S66, SonyEricsson P900, …) and at the same time provides the opportunity to implement a real-time bidirectional communication.

## 2. System approach - Overview

According to the argumentation above, this novel approach makes use of Java applications for mobile devices, especially mobile phones. These applications are called Midlets, which are to be regarded as mobile relatives of the well-known Java-applets. A drawback of this approach is the fact, that in contrast to web-browsers, these applications are not pre-installed by mobile device vendors. On the other hand, the necessity to distribute these Midlets can be regarded as a big chance to realise user-dependent applications, which meets the demand for personalisation. A further issue to be dealt with is the commissioning of the system, i.e. configuration of the application according to the KNX installation to be controlled. These challenges have been met by an integrated system approach depicted in figure 1.

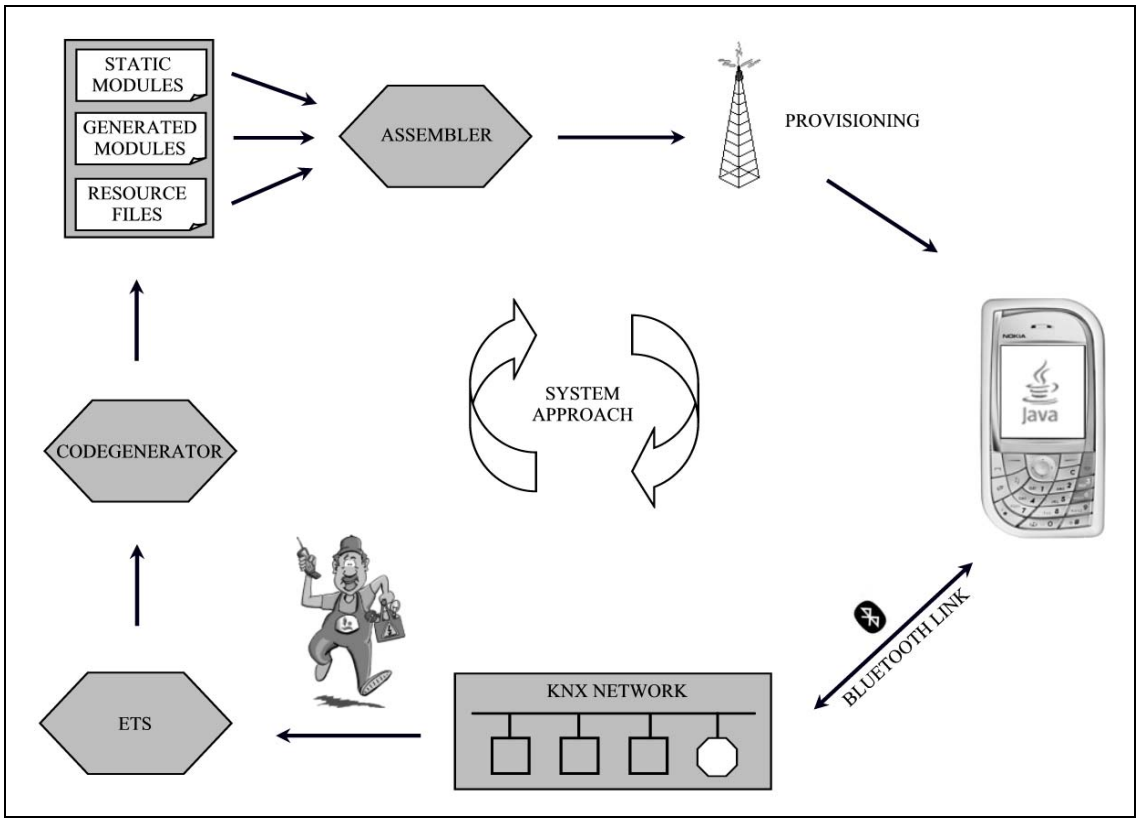


Fig. 1: System approach

This system approach is not limited to the communication path between KNX installation and mobile devices (shown on the right side), but also considers the whole supply chain for a personalised Java Midlet. The electrician defines a project in the ETS that is used by a code generator to produce an application suited for the installation (refer to chapter 4.2). The tailored application is then deployed to the customer (e.g. over-the-air provisioning by SMS, assuming a mobile phone is used). After the simple installation on the mobile device, the user has control over the whole installation.

## 3. Hardware aspects - Realisation of a Bluetooth-PEI

Considering the communication between KNX installation and mobile device, there is not only an application needed for the mobile device, but there has to be a kind of gateway towards the KNX world, in this case a Bluetooth gateway. As already mentioned above, most approaches lead to web-server implementations for the gateway functionality. This novel approach aims at realising a very thin gateway concept, more or less a kind of media coupler providing a wireless PEI. The developed module can be plugged onto a standard BCU 1 and makes use of a GoldCap and duty-cycling to meet the energy requirements. Because of its tiny functionality to just make the PEI wireless and extend it via Bluetooth to the mobile device, the hardware realisation is not too complex. A standard 8-bit microcontroller like the Atmega32 and a simple Bluetooth module supporting the serial port profile (SPP) are the main components.

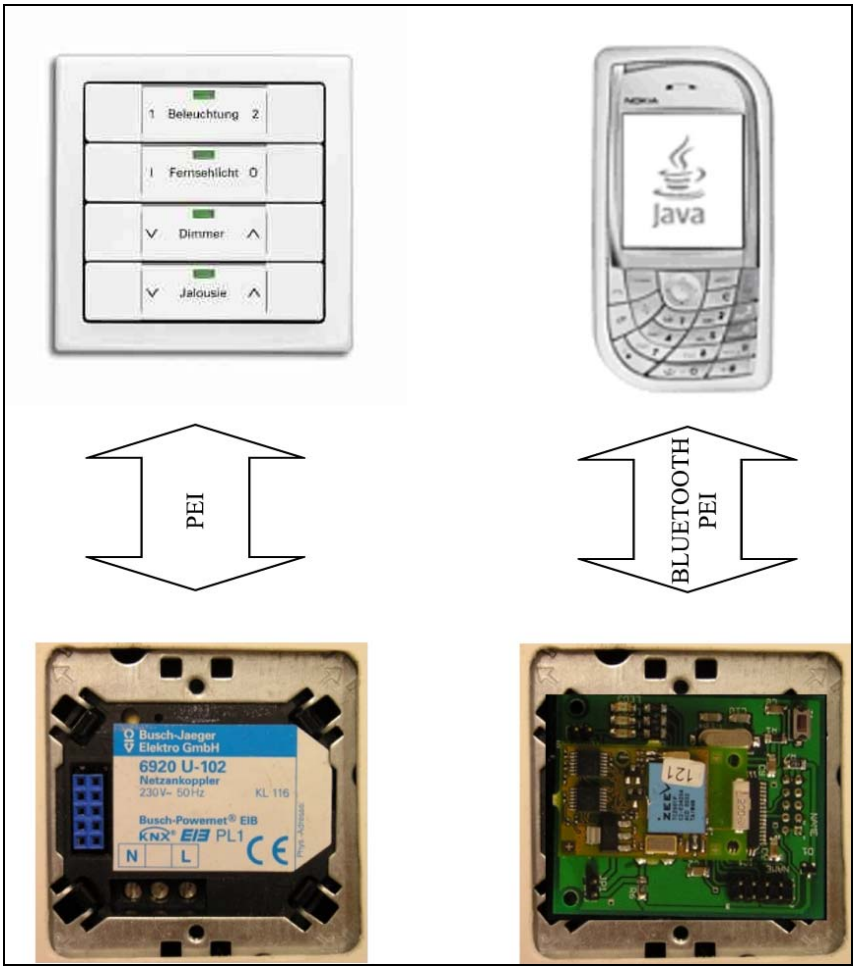


Fig. 2: Bluetooth PEI

Figure 2 displays the consequences of the wireless PEI approach. The mobile device is wirelessly plugged onto a BCU and thus behaves like a 4-fold switch or any other application module in the KNX world. Because of the fact, that KNX telegrams are generated and received by the application running on the mobile device, it becomes clear, that real-time status-consistency is automatically achieved.

## 4. Software aspects - From concept to deployment

In contrast to the simple hardware realisation, the software concept and deployment turns out to be rather challenging. As mentioned in chapter 2, the whole supply chain for the Java Midlets has to be taken into consideration. Without going too deep into detail, the next two subchapters shall give an impression on how the challenges have been met.

## 4.1. Concept and implementation of the mobile applications

The mobile applications are realised as Java Midlets to achieve a platform-independent solution. The usage of Bluetooth from these Midlets is made possible by a special API called JSR-82. Thus, it is possible to establish Bluetooth connections and transfer data wirelessly from Java Midlets. The Midlets are realised in a consequent modular manner. While one component deals with the Bluetooth access and connection management, the component shown in figure 3 deals with the application specific parts like user-interface (view), data-access (model) and the business-logic (controller), which of course includes the generation of KNX telegrams upon user request (e.g. pressing a button on the display) and the interpretation of received KNX telegrams via Bluetooth including automatic update of data representation and display content.

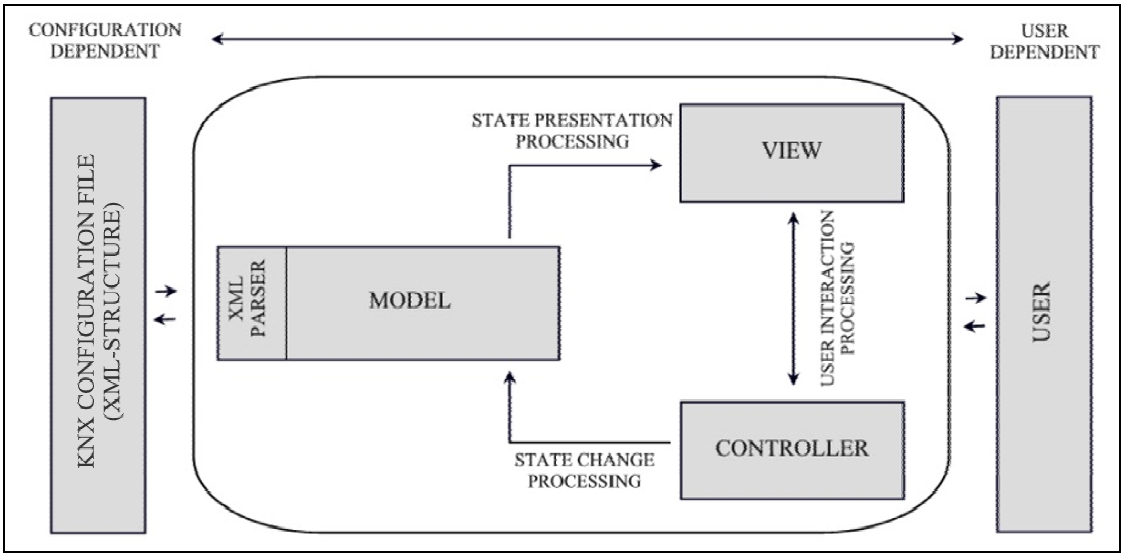


Fig. 3: Software concept according to the model-view-controller paradigm

One interesting part of the software is the integration of an XML-parser, which operates with an external XML-file (eXtensible Markup Language) that keeps the information about the KNX installation to be controlled. The distinction of code implementing the functionality and the user-interface from the data representation guarantees flexibility for future use.

## 4.2. Code generation and application deployment

In the current approach, the Java Midlets as well as the XML-files containing the information regarding the KNX installation are generated by a webservice, which takes the user's preferences regarding the user interface (colours, shapes etc.) and the ETS project information as input. These determinants for the final application are displayed in figure 4.

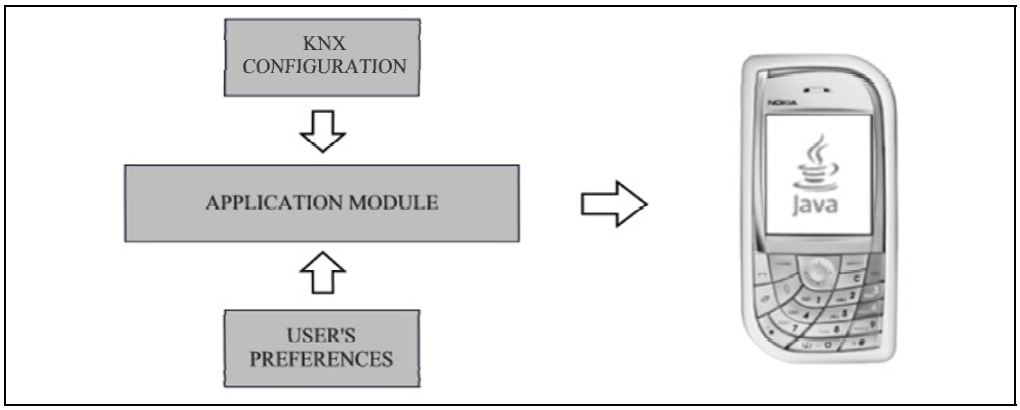


Fig. 4: Application code generation

The generation of the XML data representation is performed in two steps shown in figure 5. The ETS export file is converted into an intermediate XML structure and in a second step transformed by a XSL stylesheet (eXtensible Stylesheet Language). This step is necessary to consider the syntactic and semantic rules the electrician made use of while setting up the project.

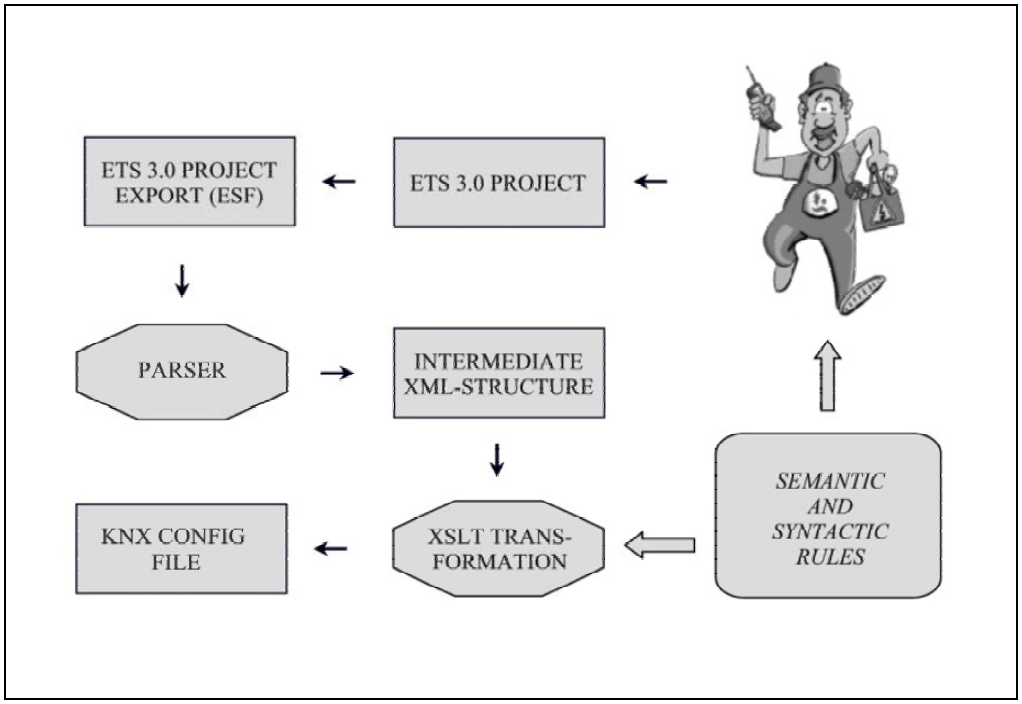


Fig. 5: Configuration file generation

The process of XML-file generation shown above has to be regarded as an intermediate solution. A standardised XML representation of KNX installation data may someday lead to an even easier commissioning of the whole communication system.

## 5. Conclusion – On the way to Ambient Intelligence

In the previous chapters, a novel approach to the integration of mobile devices, especially mobile phones, into KNX environments has been presented. The approach makes use of a very simple gateway, rather a media coupler, extending the PEI wirelessly via Bluetooth to plug a mobile device like a standard KNX application module onto a standard BCU 1. On the side of the mobile devices, mobile applications called Java Midlets are used, which can access the Bluetooth functionality by a new API called JSR-82. The system approach does not only consider the communication between KNX installation and mobile device, which inherently leads to real-time status consistency, but also puts a focus on application generation and deployment.

The modular architecture of the software allows the XML-based data representation of the KNX installation even to be received via Bluetooth at runtime. Thus, it is possible to have a tailored application (in terms of a personalised user-interface) running on your mobile phone while entering a room, containing the Bluetooth gateway that provides XML-based information regarding the installed KNX devices in the room. The Bluetooth gateway can now find and connect to your mobile phone (by Inquiry and Paging) and consequently present to you available services in the room (switching lights, controlling the blinds, …). This Ambient Intelligence scenario where the room offers its available services to the user without explicit request may be considered useful especially for complex conference rooms.

## 6. References

[1] S. Jurthe: „Aufbau eines Frameworks zur Bereitstellung modularer Bluetooth-Anwendungen auf mobilen Endgeräten“ (University Dortmund, CEI - Computer Engineering Institute, 2004)

[2] B. Wilms: „Entwicklung eines Bluetooth-Gateways für den European Installation Bus zur Kommunikation mit mobilen Endgeräten“ (University Dortmund, CEI - Computer Engineering Institute, 2004)

[3] B. Wilms: „Entwicklung eines Frameworks zur automatisierten Generierung plattform-unabhängiger Applikationsmodule für mobile Endgeräte zur Kommunikation über ein Bluetooth-EIB-Gatewaysystem“ (University Dortmund, CEI - Computer Engineering Institute, 2004)