Abstract
In this demo we will show how devices from different vendors, using different protocols, can be combined and made to work together without detailed low-level programming by the user. The small example we have chosen uses a radio-controlled power socket from one vendor and a temperature sensor from another vendor. We use these to create a remotely controlled electric kettle, which keeps the water at the point of boiling, ready to make tea at any time. We also show how we very easy can use a mobile phone for remote control and monitoring of the kettle. It is all built with a simple-to-use graphical user interface offered by the PalCom middleware, and will be modified as part of the demo.

Author Keywords
PalCom; middleware; ad-hoc composition; devices; services; pervasive systems

ACM Classification Keywords
C.2.4. Distributed Systems: Distributed applications; H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Design, Human Factors, Languages
Introduction

The PalCom middleware enables Devices to be combined across heterogeneous networks, and Services they offer can communicate despite they are not made to work together. The PalCom middleware supports this by offering mechanisms for discovery and routing between network technologies, a standardized way to exchange descriptions of services (rather than standardize the services themselves), and a combination mechanism, Assemblies, where configuration and coordination scripts are stored. Assemblies can be created with a graphical editor enabling also non-programmers to configure their devices in new ways.

We demonstrate the use of PalCom in a simple scenario, on a small set of devices, but still show the power of this approach since they are provided by different vendors, using different protocols for communication, over different network technologies, but can still be combined transparently using our graphical editor. This being a scaled down demo example we would like to point out that the PalCom middleware is used for much larger and demanding applications - in healthcare: support for advanced homecare, and - in automation: building a mobile manipulating robot.

PalCom overview

PalCom is a middleware framework [2][3] used to combine the services offered by devices in an easy and flexible manner [1]. By doing this, new functionality can be created by coordinating already existing services in new formations. In PalCom, all devices are represented as PalCom Devices. Similarly, all functionality provided by such devices is presented in the form of PalCom Services. PalCom enables a homogeneous representation of system components, hiding complexity of systems using heterogeneous networks. Once created, PalCom components can be used and reused in any number of solutions. PalCom components are selected and coordinated (assembled) into PalCom systems by defining PalCom Assemblies. Assemblies have a configuration part, which defines what devices and services to include, and a coordination part, which is a simple script to define the coordinated behavior of services.

The ability to offer transparent combination of devices across heterogeneous networks is enabled by a network independent addressing mechanism. This means that devices can move between networks and still be attached to its system(s), and that Assemblies, which stores bindings to devices, can be moved between devices in the PalCom network and still connect to the correct devices. This creates a unique flexibility in moving devices and functionality in a network without reconfiguration. PalCom is implemented in Java [4] and runs on most computers including Mac OS-X, Windows, Linux/Unix, and Android. A partial implementation in C makes it possible to include also smaller devices, although with limited functionality. The PalCom toolset consists of:

- PalcomBrowser – used for browsing Devices and Services, interacting with them and creating Assemblies encapsulating configurations using a graphical editor.
- TheThing – a generic PalCom device, which acts as a virtual machine for Services and Assemblies.
- palcom-lib – a collection of Java classes used for development of new Services.
Demo-System Overview
In the small example we have chosen for the demo, an electric kettle, a power socket that can be remotely controlled, and a temperature sensor connected to a computer. We use a laptop and two mobile phones with PalCom tools for modifying the set-up and interaction. This is only a starting point for the demo. During the demo, the system can be modified in real-time to meet the needs of the visitors, or a completely new system can be drafted from scratch.

Controlling Computer: TheThing
To this computer a TellStick USB dongle, which is used to wirelessly control the state of the wall power-plug, has been plugged in. TheThing is loaded with a service that understands the TellStick protocol. When the TellStick device is discovered by the TellStick discovery mechanism, a new PalCom service is created to represent the power-plug, to communicate its state, and to control it from PalCom. In Figure 1 the plug for the kettle is represented by the service labeled “Boiler” providing commands to turn the power of the wall plug and thus the kettle, on or off. Similarly, a thermometer inside the kettle, attached via 1-Wire USB dongle, and a matching PalCom service, provides readings accessible through the “Temp” service. This service monitors the current temperature by periodically sending a PalCom command to all services that have connected to it.
One additional service, “TempGuard” offers the function of: given a temperature as input, provide an output command when the given temperature is below (turn on kettle) or above (turn off kettle) a certain value.

**Laptop: Assembling the System with the PalcomBrowser**

The interactive editor of the PalcomBrowser is used to put together the Assembly, “OnOffSwitch” which uses the three existing services to together work as a thermostat as depicted in the bubble. To provide a simple on/off interface to the system, the assembly defines a synthesized service. This service takes input commands to either activate or deactivate itself. The PalcomBrowser can also be used to interact with discovered services (e.g. turn power on and off), run the created Assembly for test purposes, and again interact with the services it offers.

**Controlling Computer: running the Assembly.**

The created assembly can now be installed in TheThing for more of a “production” setting. With the computer setup to auto-start TheThing and thus the three services and assemblies it manages, makes the functionality available at all times, “blending into the walls” as other sort of wiring.

**Smartphone: Turning the Kettle On with an SMS**

PalCom Assemblies can build on each other, which we use to create a slightly more complex system. An Android smartphone with TheAndroidThing offers, in particular, the "SMS Service" which can send and receive SMSs. It forwards received SMSs in the form of PalCom commands. Combined with "SMS Parser", on-or off-commands can be produced from SMSs. Finally, the second assembly “SmsSwitch” assembles the services and the sub-system of “OnOffSwitch” into a complete system. With everything running, whenever the phone receives an SMS containing the text “on”, “OnOffSwitch” will be activated and the electric kettle will turn on and maintain a boiling temperature. When it receives “off” if will deactivate "OnOffSwitch" which in its turn will switch the kettle off.

**Conclusions**

In this demo we have shown how the PalCom tools enables the combination of self-describing services, using different protocols, distributed across different devices. The high-level approach of the architecture effectively hides the complexities otherwise associated with heterogeneous network solutions. Interactive tools like TheThing and PalcomBrowser simplifies the configuration and reconfiguration of systems, making them easy to extend and/or alter.

**References**


