

3PC/MarNET Pervasive Presenter

Marcus Handte*, Stephan Urbanski*, Christian Becker*, Patrick Reinhardt+, Michael Engel+,
Matthew Smith+

* Institute of Parallel and Distributed Systems
Universität Stuttgart, Germany
firstname.lastname@informatik.uni-stuttgart.de

+ Dept. of Mathematics & Computer Science
University of Marburg, Germany
{reinharp,engel,matthew}@informatik.uni-marburg.de

Abstract

This demonstration shows the main features of PCOM [3] and BASE [2], our component system and middleware for pervasive applications. To do this, we present an application that leverages the resources present in the environment. Through the configuration mechanisms and algorithms provided by PCOM, the application can be automatically adapted without user intervention or explicit programming. To emulate various network properties and scenarios (e.g., packet loss and disconnects), we use the MarNET Emulation Testbed [4].

1. Motivation

Pervasive Computing envisions *distraction-free* support for everyday tasks by leveraging the functionality that is present in the environment of a user. This vision requires adequate software infrastructures that provide developers with the necessary mechanisms to create *self-configuring* adaptive applications. To this end, we have developed the component system PCOM [3] on top of our middleware BASE [2]. In this demonstration, we present a PCOM application that shows the main features of these systems.

2. Application

The Pervasive Presenter application displays PowerPoint™ presentations. Using a

PDA, a user can start the application by invoking the anchor component on his device as shown on the left side of Figure 1.

The anchor (see Figure 1, *Control*) defines a number of dependencies in its PCOM contract (see Figure 1, *Infrared*, *Presenter*, *Filesystem*). As long as the anchor is executed, the PCOM component container ensures that dependencies are resolved by binding adequate components to them. Such components can be distributed across the devices in the environment. Using the device discovery mechanisms of BASE and the contract matching and (re-) configuration algorithms of PCOM, the container can automatically find such components without relying on manual configuration or explicit programming.

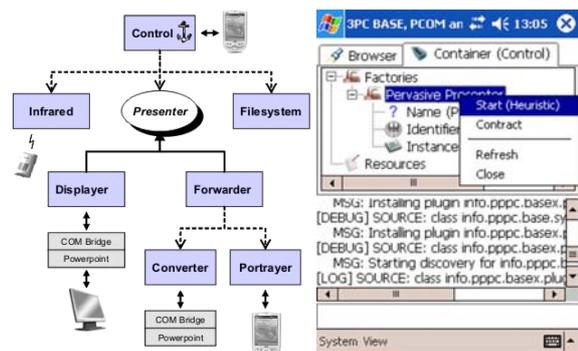


Figure 1: Application Structure and PDA GUI

In our exemplary environment, there is one possibility to bind an *Infrared* component and one *Filesystem* component. Additionally, we provide two different implementations for the *Presenter* component. One component (see

Figure 1, *Displayer*) uses Microsoft PowerPoint™ directly to show the presentation, thus it can solely run on heavy-weight devices like laptops. The other one (see Figure 1, *Forwarder*) uses PowerPoint™ as a converter and it relies on another component (*Portrayer*) that displays the converted slide as an image. In contrast to the *Displayer*, the *Portrayer* can present slides on devices that do not have the capability to run PowerPoint.

In order to switch between the different configurations, the application relies on IR beacons that are received through the *Infrared* component. By pointing the PDA to the beacon representing the *Displayer* configuration, the application anchor will change its contract in such a way that only *Displayer* components are selected for the *Presenter* dependency. If the PDA points to the *Portrayer* beacon, the inverse happens. Thus, IR beacons can be used to create a follow-me presentation application that uses the output capabilities in the vicinity of the user. Thereby, the user does not have to reason about the configuration that is required in order use the output device.

3. Technical Setup

For the Pervasive Presenter application, we use two Pocket PCs (*Control/Infrared* and *Portrayer*) and one laptop (*Filesystem, Converter, Forwarder* and *Displayer*). Furthermore, we use two IR beacons to switch between the configurations.

In order to emulate various network parameters (e.g., disconnects and packet loss), we use the MarNET Emulation Testbed [5]. This network emulator runs on a dedicated laptop. It provides a graphical front end that visualizes the network environment.

The Pocket PCs and the laptop that run the presenter application connect to the emulation laptop via WLAN. All traffic between the devices running the application is tunneled through the emulation laptop. The emulation

laptop runs a number of virtual machines that form the network environment. Three of these virtual machines represent the existing devices. Further virtual machines are used as routers between them.

4. Contribution

The contribution of this demonstration is twofold. First, we present an important part of the software stack developed by the Peer-to-Peer Pervasive Computing project [1] over the last three years. With our focus on automatic configuration and adaptation, we hope to facilitate a discussion and a deeper understanding for the benefits and limitation of the automation provided by the PCOM component container. Secondly, we show how emulation tools like the MarNET Emulation Testbed [4] can be used to emulate various network properties by integrating virtual nodes that only exist in the emulation scenario and real devices such as PDAs or laptops.

References

- [1] Peer-to-Peer Pervasive Computing Project Homepage, <http://www.3pc.info>
- [2] Becker, C., Schiele, G., Gubbels, H., Rothermel, K., „BASE – A Micro-broker-based Middleware for Pervasive Computing”, IEEE International Conference on Pervasive Computing and Communications, March 2003, pp. 443-451
- [3] Becker, C., Handte, M., Schiele, G., Rothermel, K., „PCOM – A Component System for Pervasive Computing“, IEEE International Conference on Pervasive Computing and Communications, March 2004, pp. 67-77
- [4] MarNET project Homepage, <http://ds.mathematik.uni-marburg.de/de/research/project.php?name=routing>
- [5] O. Battenfeld, M. Smith, P. Reinhardt, T. Friese, and B. Freisleben. A modular routing architecture for hot swappable mobile ad hoc routing algorithms. In Proc. of the Second International Conference on Embedded Software and Systems, Springer-Verlag, 2005 (to appear).

Acknowledgements

This work is partially funded by German Research Foundation (DFG) Priority Programme 1140 Middleware for Self-organizing Infrastructures in Networked Mobile Systems.

Biography



Marcus Handte is a researcher of the Distributed Systems Research Group at the Institute of Parallel and Distributed Systems at the University of Stuttgart. He received a M.S. degree in Computer Science from the Georgia Institute of Technology in 2002 and a diploma in computer science from the University of Stuttgart in 2003. Currently, he is pursuing his PhD with a focus on the area of system support for pervasive applications.



Christian Becker received his diploma in computer science from the University of Kaiserslautern in 1996 and his PhD from the Universität Frankfurt in 2001. Since 2001 he is working as senior researcher and lecturer at the Institute for Parallel and Distributed Systems at the University of Stuttgart. His research interests are middleware platforms and context-aware computing.



Stephan Urbanski is a researcher at the Institute of Parallel and Distributed Systems, Department Distributed Systems at the University of Stuttgart. He obtained his M.S. in Computer Science from the Georgia Institute of Technology, Atlanta, Georgia in 2003 and his diploma from the University of Stuttgart, Germany in 2004. His research interests include pervasive middleware infrastructures, process oriented application development and mobile computing.



Patrick Reinhardt obtained his diploma in computer science in 2005 from the University of Marburg, Germany. Since then, he is working on mobile ad-hoc networks in the Distributed Systems Group at the University of Marburg, pursuing a Ph.D. in computer science.



Michael Engel is working as a senior researcher in the Distributed Systems Group of the University of Marburg since 2002. He received his diplomas in mathematics in 1999 and computer science in 2002 at the University of Siegen. In 2005, he obtained his PhD at the University of Marburg. His major research interests are operating systems, computer networks and aspect-oriented software development.



Matthew Smith is working as a researcher in the Distributed Systems Group of the University of Marburg since 2003. He received his diploma in computer science in 2003 at the University of Siegen. Currently, he is pursuing his PhD with a focus on Ad-hoc Grid Security.

