The Energy Aware Smart Home

Marco Jahn, Marc Jentsch, Christian R. Prause, Ferry Pramudianto, Amro Al-Akkad, René Reiners
Fraunhofer Institute for Applied Information Technology FIT

Schloss Birlinghoven, 53754 Sankt Augustin, Germany
{marco.jahn, marc.jentsch, christian.prause, ferry.pramudianto, amro.al-akkad, rene.reiners}@fit.fraunhofer.de

***Abstract*—In this paper, we present a novel smart home system integrating energy efficiency features. The smart home application is built on top of Hydra, a middleware framework that facilitates the intelligent communication of heterogeneous embedded devices through an overlay P2P network. We interconnect common devices available in private households and integrate wireless power metering plugs to gain access to energy consumption data. These data are used for monitoring and analyzing consumed energy on device level in near real-time. Further, transparent information about the energy usage can be used to efficiently program and control home appliances depending on various factors, e.g. the electricity price. Making more and more data available to end-users, brings with it further challenges in the area of user interfaces. Hence, we complete the smart home system by intuitive user interfaces presenting energy consumption data in meaningful contexts and allowing end users to interact with their environment. We argue, that the combination of both, a technically sophisticated smart home application and at the same time transparent, intuitive user interfaces showing information regarding the energy usage, e.g. energy price, energy source, standby consumption etc., has the potential to bring the vision of the energy efficient smart home within reach.**

***Keywords*-smart home; middleware; energy efficiency; user interaction;**

I. Introduction

Global warming — and its desastrous environmental and economic effects — is considered one of the major challenges that mankind will face during this century. The problem is mainly attributed to CO_2 emissions that, for example, arise from the generation of electricity from fossil fuels. One way to reduce emissions of CO_2 is therefore to reduce the overall consumption of electricity in industry and the private sector; especially the latter one is what several national and international initiatives aim at.

Home owners, of course, have themselves a high interest in reducing energy consumption because energy is an important cost factor. Here usage awareness alone has the potential to reduce consumption by 15% in private households [19]. However, standard electricity meters that are widely deployed in homes today, and the suppliers' analog billing systems based on yearly accounting periods, lack the feedback capabilities that are necessary to increase energy awareness and positively affect customers' behavior [4].

This is where smart metering offers alleviation. Smart metering shortens the feedback time from consumption of the energy to user billing considerably, and only by this enables energy awareness. The young field is rapidly evolving through the research efforts of several large electricity companies as well as publicly subsidized research projects.

Yet current solutions of smart metering are proprietary and generally not generic or flexible. Even an agreement on a common standard on smart metering technology, protocols etc. does not seem to be in sight. Thus, to bring to the users the benefits of smart metering, it is worth researching generic solutions, that are linked to the user rather than to the energy provider and are independent of any proprietary hard- and software. Intelligent smart home environments seem to be a promising basis for incorporating energy efficiency features in private households. Besides pure monitoring, such smart homes already provide the infrastructure to use e.g. energy pricing information to control devices.

In this paper, we present a system that integrates into the smart home environment energy efficiency features, made accessible via innovative approaches towards user interaction. By using the Hydra[1] middleware framework for developing the smart home environment, our approach is independent of any provider specific infrastructure. Hydra is a generic middleware for developing networked embedded systems [6]. It allows developers to incorporate heterogeneous physical devices into their applications by offering web service interfaces for controlling any type of physical device irrespective of its network technology such as Ethernet, Bluetooth, RF, ZigBee, RFID, WiFi, etc. As generic middleware framework, Hydra offers a high degree of flexibility, making it applicable in various domains like home automation or e-healthcare [5] [14] [8]. The core components of the Hydra middleware will be released as Open Source [12]. This enables a wide-spread adoption of the middleware for personal and commercial purposes alike. Hydra as basis for interconnecting household appliances lays the ground for the presented energy aware smart home system. To integrate energy awareness into this system, we use Plogg[2] wireless plugs, that are capable of accessing

[1] http://www.hydramiddleware.eu/
[2] http://www.plogginternational.com/

978-1-4244-6949-9/10/$26.00

energy consumption data. We realize not only monitoring but also control functionality with innovative approaches to user interaction like UbiLense [13]. We show that Hydra as a generic middleware framework serves well as for integrating energy awareness into the smart home and that the energy efficient smart home does not solely depend on energy suppliers or device manufacturers, but can be achieved on the userside in private households.

In the following we provide an overview of related work in the relevant research areas as well as industrial efforts in section 2. In section 3 we describe our energy aware smart home application from the user's perspective and point out the main features. Section 4 deals with the technical architecture, introducing Hydra, Ploggs, and elaborating on the integrated system. Section 5 describes our approach to user interaction - UbiLense - and we conclude in section 6.

II. Related Work

Several related research as well as industrial projects exist within the main topics addressed in this paper, which are middleware for embedded systems, energy awareness and monitoring and mobile interaction with the real world.
Several projects related to middleware for networked embedded systems exist. For example, the AMIGO [16] project aims at developing an open, standardized, interoperable middleware and intelligent user services for networked home environments. SOCRADES [9] similar to Hydra, explores a service oriented middleware approach to integrating device level information into business processes. It also provides features like eventing, management and service discovery. The applicability of such generic middleware in the area of energy aware home automation has yet to be researched.
Besides such generic middleware approaches, other projects focus on researching specific energy aware solutions. For example, the eDiana[3] project aims at increasing energy efficiency of embedded devices by developing a reference model-based architecture to foster the integration of existing infrastructure into the power grid. The AIM [3] project puts efforts in developing technologies for managing energy consumption in domestic environments in real-time. Residential users should be able to administer their home networks while functionalities are exposed as services to outside networks via gateways offering functions for policy management, device discovery, and proactive configuration. Both, eDiana and AIM provide rather specific solutions for outfitting households with energy awareness features. In contrast, by using Hydra as a generic middleware, we are able to combine energy efficiency with any kind of other smart home devices and services.
Apart from research projects, the industry is currently focussing on providing a wide range of energy monitoring solutions like Google PowerMeter[4], Greenbox[5] or The Power Tab[6]. Common industrial solutions are proprietary and depend on exisiting smart metering tools, while most of them are even unable to monitor consumption on device level. Our approach to energy consumption monitoring can be applied in any kind of environment, regardless of any proprietary protocols, devices etc.
The Magic Lens metaphor which we adapt for our interaction technique was first described by Bier et al. [2]. It is a transparent visual filter which reveals hidden information about an object by enhancing the presentation of the object with data of interest. The transparent filter has to be put over the object to become an See-Through Interface. Rekimoto presented an implementation of this concept using a small LCD-TV screen and a miniature CCD-camera [15]. The setup is used to present textual information on objects like a paper calendar when pointing the screen at them. Wagner et al. enhanced Rekimoto's approach to work on a camera-enhanced PDA [18]. In an indoor navigation scenario, visual markers are attached to decision points on the route. Users can point their PDA to the marker and are provided with the information where to go next. They use 2D barcodes for object recognition. A markerless approach is presented by Rohs et al. [17]. For using the Magic Lens on a paper map, patches of the map are pre-stored in a database. The camera image showing the map is then compared against these patches.

III. Energy Aware Smart Home

The presented application aims at integrating energy efficiency into a smart home infrastructure, providing intuitive user interfaces for monitoring and controlling the smart environment. The user can interact with the system via both, stationary and mobile interfaces. Cumulative and comparative views on devices and energy consumption are presented on a large scale display like a computer screen, TV, etc. Figure 1 shows this user interface providing easy-to-compare per device information on the right side. For each device it displays the current consumption in watts, costs per hour, and costs projected over one year using an adjustable average per-day usage time. Costs are calculated taking into account the electricity price, which depends on the daytime. To demonstrate the effects of variable energy prices, the daytime can be changed manually using the oval slider on the top left of the interface. Cumulated consumption and cost data of all devices are shown below. These values, as well as the per-device values are updated every second.

On their mobile devices, users can directly access appliances using UbiLense. Figure 2 shows an example of the UbiLense interaction concept. It recognizes objects using

[3]http://www.artemis-ediana.eu/index.php

[4]http://www.google.org/powermeter/

[5]http://getgreenbox.com/

[6]http://www.energy-aware.com/our-products/ihd/

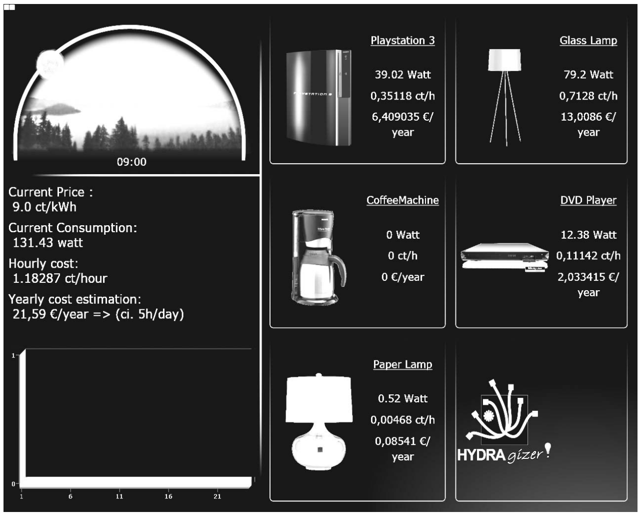

Figure 1. Stationary Monitoring Device

Figure 2. Mobile Device - UbiLense

image processing methods and displays energy consumption information about that device. Further, the application supports seamless interaction between the different types of user interfaces. For example, once a device is recognized on the mobile phone, the user can push it to the monitoring screen to compare its consumption with other devices; if he is no longer interested in this device, he can remove it.

In future smart homes, energy efficiency will be a major issue, when considering advancements in the area of smart metering and smart grids [7] . Energy providers will have to react to liberalized markets, micro energy generation and resulting user requirements. For example, time-of-use and real-time pricing may help both, consumers to save energy costs in households and energy providers to improve load management. [4]. Further, if users had access to transparent information about which energy source (wind power, nuclear power etc.) is in use at a certain time, these data could also be taken into account. For example, users would be rewarded when using green energy and accordingly the price might rise for using black or nuclear energy. In the household it all comes down to using this information, meaningfully processing and visualizing it. We believe that an energy aware smart home application has to find a balance between supporting the user in saving energy and at the same time not decreasing convenience. Thus, besides pure monitoring of energy consumption, we provide novel control functions for increasing energy efficiency in households. On the stationary control device, users can manually change the energy price and are continuously informed by the system regarding changing energy costs. To demonstrate features of future energy usage, we integrate a virtual washing machine into the smart home application, i.e. a PC renders a video of a running washing machine. Users can individually program this virtual washing machine by setting a limit for the energy price. If the energy price falls below this limit, the washing machine will start.

IV. System Architecture

We build our system on top of the generic Hydra middleware framework. Hydra aims at supporting the development of networked embedded systems, simplifying the interconnection of heterogeneous devices communicating over different protocols. We use the Hydra middleware to build our smart home network consisting of various home appliances. Further, we connect each device to a Plogg to be able to use the energy consumption data in the smart home application and seamlessly integrate the UbiLense user interaction concept into the system.

A. Hydra Middleware

The Hydra middleware framework facilitates developers to build efficiently scalable, embedded systems while offering web service interfaces for controlling any kind of device irrespective of its network technology. Implementing a service oriented architecture, Hydra comprises a set of sophisticated components (called managers) taking care of P2P communication between devices, security, event handling, context awareness etc. Each manager is devoted to a defined task inside the middleware and exposes its services to be consumed by other components, following the principle of separation of concerns.

Figure 3 provides an overview of the Hydra architecture, displaying all managers that are part of the middleware, some of which are essential (e.g. Network Manager) and others which provide optional functionality (e.g. Context Manager or Storage Manager). Each manager has a clearly-defined role, offering a set of services to be used by other managers or application level components. Further, as Hydra aims at supporting the development of distributed systems,

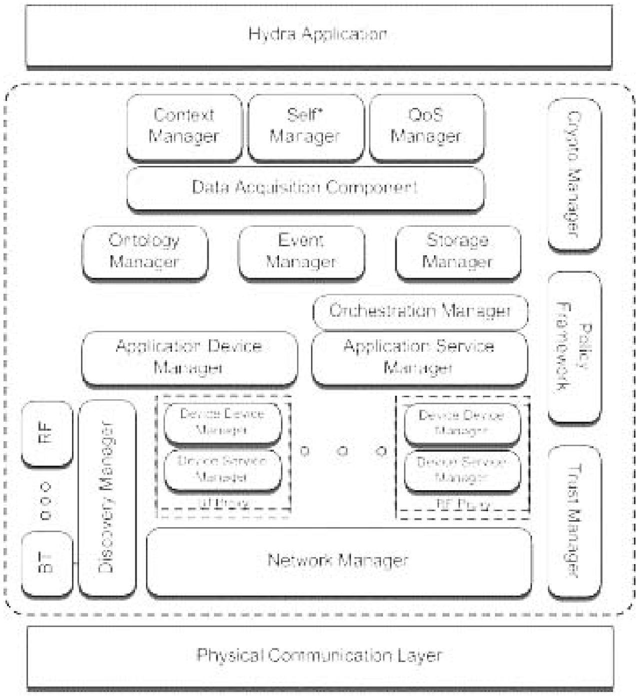

Figure 3. Hydra Architecture

managers can be deployed on different hosts, communicating via web services. In consequence Hydra supports the development of various kinds of applications, from simply connecting two computers to full-fledged pervasive environments supporting security, storage and context awareness distributed over many heterogeneous devices, depending on the domain-specific requirements.

Hydra distinguishes two types of devices: Those that are able to host parts of the middleware, and those that are not. The minimum configuration includes only the Network Manager, enabling P2P communication among devices. Devices that are not able to host a Network Manager, due to limited resources or the like, can be connected to the Hydra network by software proxies running on more powerful devices. Examples of resource restricted devices that the Network Manager has been successully deployed on are the Playstation 3 and the Android G1 mobile phone [1].
The Network Manager enables network communication among devices inside a Hydra network. It creates an overlay P2P network that implements SOAP Tunneling as transport mechanism for web service calls [10], allowing direct communication among all devices inside a Hydra network, no matter if they appear behind a firewall or NAT (Network Address Translator). Hydra uses a custom addressing mechanism that assigns unique identifiers (Hydra-ID, HID) to services, allowing devices to transparently publish and use services anytime anywhere regardless of network boundaries or fixed service endpoints. If a device wants to consume a service of another device, the Network Managers of both devices take care of routing the web service calls, using the services' HIDs.
Besides the Network Manager, which is the essential Hydra component, we employ the Event Manager. For a smart home application it is essential to be modular, extensible and provide low coupling of components, as set-ups can change when devices are removed or new devices are added to the environment. Therefore, we deploy the Hydra Event Manager, which implements a publish/subscribe mechanism for services. For example, if a device is unplugged from its Plogg, an event can be fired, telling the monitoring interface not to display this device any longer. And, talking about extensibility, any device that would be interested in such an event can simply subscribe for it. The Event Manager handles all subscriptions and is responsible for publishing events, again via a Network Manager, compliant to the Hydra communication model.
The Hydra reference implementation is built for OSGi[7] environments. OSGi adheres to the principles of service- and component-oriented programming, providing a java-based modular service platform. Components (called bundles) can be installed, started and stopped at runtime, not requiring a reboot of the whole environment. Each bundle publishes services that can be looked up and consumed by other bundles. Consequently, Hydra managers are available as OSGi bundles that can be plugged together on demand. Further, each manager publishes a SOAP web service interface to facilitate remote communication among components.

B. *Plogg Integration*

Any serious energy consumption monitoring or control tool should be able to provide consumption data on device level, to provide maximum transparency. We use Plogg wireless smart meter plugs to receive energy consumption data of plugged-in devices. Ploggs can be accessed wirelessly either via ZigBee or Bluetooth and provide information like currently consumed watts, kWh generated, kWh consumed, frequency and many more. To provide users with direct feedback on their current consumption, we mainly make use of the consumption information of each device.
To integrate the Plogg functionality with Hydra we developed an OSGi bundle that is capable of exposing Plogg services inside the OSGi environment and as SOAP web services. Currently, the Plogg SDK is available only as Windows DLL. We enable Java-access to the Plogg library by developing a JNI (Java Native Interface) bridge that wraps the DLL allowing transparent Plogg access inside an OSGi environment.

C. *Integrated System Architecture*

Figure 4 provides an overview of the integrated system architecture, showing Hydra managers as well as application

[7]http://www.osgi.org/

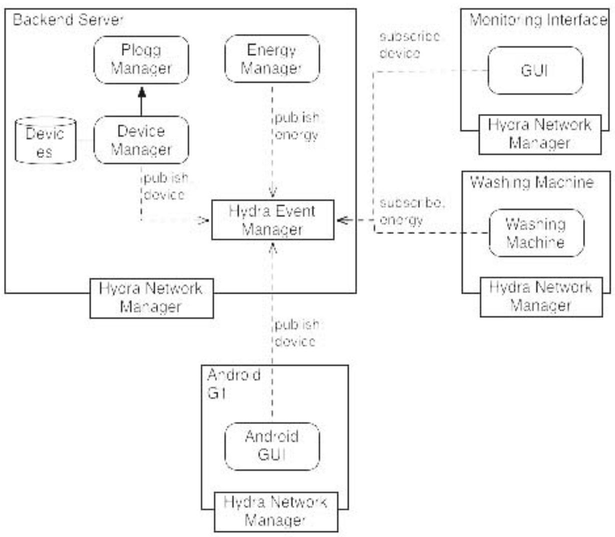

Figure 4. System Architecture

components. We distributed our application over 4 different devices, all of which are capable of hosting parts of the Hydra middleware. The Backend PC is Desktop Computer, hosting the basic infrastructure like database, Plogg control etc. Then we have two Tablet PCs, one representing the stationary monitoring device, the other representing a controllable washing machine. To provide mobile access to the application via UbiLense we employ the Android G1 mobile phone. All devices host a Hydra Network Manager enabling transparent communication. The Backend PC, Monitoring Device and Washing Machine run Eclipse Equinox[8] OSGi to deploy Hydra bundles and custom application components like the Washing Machine control logic. On the Android G1we use a special version of the Network Manager that has been optimized for mobile devices. The integrated smart home system architecture comprises the following main conceptual tasks:

1) Device and Plogg Management: The Device Manager administers information about devices that are connected to the Hydra network and/or plugged into Ploggs. It knows which device is plugged into which Plogg and constantly polls all Ploggs for their current consumption.

2) Energy Management: The Energy Manager represents a component that may be installed by energy providers into the smart home application, acting as an interface between provider and consumer. In the future, when information may be sent from providers to end users, such a component could inform the smart home application about changing energy prices, peak times etc. In our smart home application the Energy Manager maintains electricity prices depending on the daytime. This can be changed by the user, to show the effect on energy consumption costs and to demonstrate energy efficient smart home functionality.

3) Network Management: In Hydra networks all communication is routed through the Network Manager to be independent of any network boundary restrictions. Thus we install a Network Manager on any of our devices. For reasons of clarity, figure 4 does not visualize communication from one Network Manager to another.

4) Event Management: The Hydra Event Manager maintains a list of event subscribers that indicate their interest in certain events. Further, it can be called by event publishers to fire events. We currently have two categories of events in our smart home application: device-related and energy-related. The Energy Manager publishes events, informing subscribers that the energy price changed and sends the new price value. The Washing Machine, registered for such events will be informed, once the energy price changes, so e.g. it can start if the energy price has fallen below a certain limit. Device-related events are published by the Device Manager as well as by the Android GUI. As the Device Manager is responsible for keeping connections to the Ploggs, it can inform other components if devices are unplugged or more generally, if a Plogg does not deliver any more consumption data. The monitoring interface is interested in such an event and will remove the inactive device from the current view. By using the Android G1, users can also manually add or remove devices from the monitoring screen. If users choose to do so, the respective event is fired and the monitoring interface is informed to remove that device.

V. User Interaction

Currently, power consumption values are hidden from the customer. She just gets an invoice about the overall consumption of the household at the end of the year, but is never aware which device contributes to it to which amount. Fluctuation of power consumption of a single device and different average runtimes makes direct comparison of devices difficult. Hence, energy consumption is a rather complex topic for the end user. Therefore, it is important to give customers access to energy consumption information in an easy and intuitive way. In the following, we present an interaction technique for gaining this information.

A. UbiLense

The most direct way of receiving information about real-world objects is by getting them from the object itself. For this, the object has to be augmented with information. As most objects cannot provide this information by nature, a mediating device is needed which makes the information virtually visible for users.

We investigated one possibility of augmenting reality with the UbiLense prototype [13]. We used a mobile phone as Magic Lens, meaning the user sees reality by looking

[8]http://www.eclipse.org/equinox/

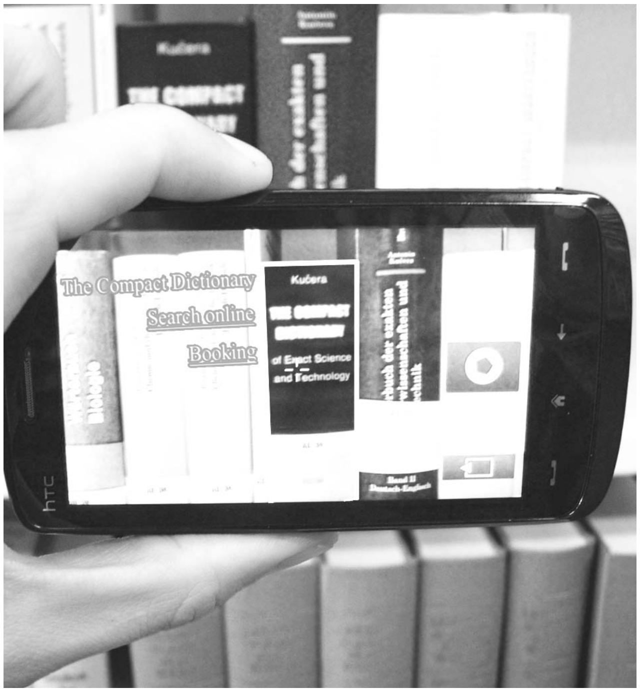

Figure 5. UbiLense visualizes the virtual services offered by a book in a library

through the camera picture [2]. The camera image shown on the device's screen is used to provide virtual services on real-world objects in form of a video overlay. The augmented object is highlighted on the screen. Figure 5 shows the application in a library scenario.

We implemented this concept using an Android based mobile phone and an image recognition server for detecting objects. The camera image of the mobile phone is streamed to the image recognition server. The server compares the images to a pre-stored set of images. It recognizes ID and position of objects. The algorithm uses a FAST corner detection, stripped down SIFT descriptors and a scalable vocabulary tree, making it fast and robust to distortion. The database contains information about the object which is sent back to the mobile phone together with the ID and position coordinates. Finally, the mobile phone can display the information on its video overlay.

For the moment, we only use the capability of UbiLense to display information. The possibility to interact with objects remotely via mobile phone will be investigated in future. The first step must be to provide people with energy consumption knowledge. Only after establishing the knowledge, people are able to react accordingly.

In the smart home scenario, the information people need is how much energy a single device consumes. For that, the image recognition server stores which device is plugged into which power Plogg. After responding this information to the mobile phone, it can request the power Plogg of the object it

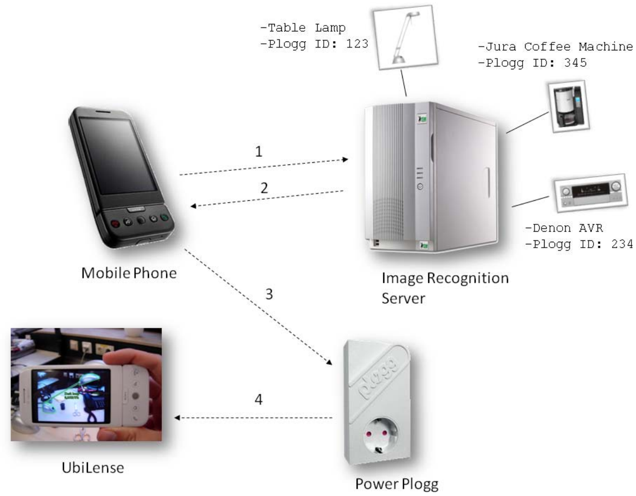

Figure 6. UbiLense architecture: Mobile phone sends camera image to image recognition server (1). Server returns name and associated Plogg ID of the recognized object (2). Mobile phone requests current energy consumption from associated Plogg (3) and visualizes (4).

is currently aiming at. The energy consumption information is finally displayed as video overlay. Figure 6 shows the system architecture.

The advantage of the UbiLense interaction technique is its intuitive and easy handling. Instead of searching through lists and navigating through menus, the user just aims at an object to select it and get immediately gets the desired information. Furthermore, the user does not have to switch his attention between his mobile device and the real world object. Switching attention would result in higher cognitive load and would make the system harder to handle [11]. By using the augmented reality approach, virtual information and reality are melted, so that switching attention is not necessary.

Pure energy consumption values in Watts are hard to conceive. Most people do not have a conception how much energy 300 Watts is. In comparison, nearly everyone can judge the value of money, e.g. 300 Euros (about 415 USD). To make the energy consumption value more conceivable, we convert Watts into the Euro equivalent. We assume an average runtime per day for each device and extrapolate the current power consumption with the current energy price to the costs in Euro per year. For example, a 40 Watt Lamp that is running for 4 hours per day costs at an energy price of 15ct/kWh:

$$\frac{40W/h \times 5h \times 365days \times 0,15Euro/kWh}{1000}$$

$$= 10,95Euro/year$$

VI. Conclusion and Future Work

Increasing the energy consumption awareness in every household is an important step to make the user able to manage his energy consumption. We have brought this concept even one step further by allowing users to observe not only the overall house hold consumption but also each device's consumption. Thus users are able to learn the energy profile of each device and to identify the devices that consume most power at home. Based on this knowledge, users have the possibility to develop better strategies for saving energy costs. Further, our system considers possible future changes in the energy market demonstrating novel functionalities for energy aware smart homes. For example, energy providers' systems could communicate directly with the smart home applications announcing special price offers during off-peak times. Users could then configure their smart devices to respond to these offers. As smart homes become even more smart, systems could learn over the time and calculate the most efficient ways to configure the home appliance or to provide users with recommendations on how to save energy. Besides the technical challenges, it is necessary to keep in mind the users' requirements. A smart home application has to be developed in a user centric way and must not be purely technology-driven. It is a thin line between an effective, user-supporting home automation system and an annoying, overly intrusive one. Now, after having a running prototype we will shift focus to user evaluations, to gain deeper knowledge on how to design energy efficient smart homes.

We also applied novel interaction techniques, which allow users to use their mobile phones as magic lenses to view the energy consumption of their appliances just by pointing gestures. When users require more details or when they like to compare energy consumption between devices, they can easily transfer the information to a larger display such as a TV. Users are also able to control the appliances such as turn on, off, start washing, play movie etc. This seamless communication among devices allows users to interact with the appliances using various kinds of device types.

We developed our system on top of the Hydra framework which, makes it very flexible and extensible. We were able to integrate without any difficulties the Plogg functionality into the Hydra environment and connect our backend application with mobile as well as stationary frontends. Benefiting from Hydra's flexible architecture, we can easily switch from a Plogg to another energy measurement device provider by replacing only one dedicated OSGi bundle in the environment. This gives us a great advantage over any vendor specific smart metering solution. Hydra even supports the integration of heterogeneous devices so we could use different measurement devices with different communication protocols from vendors as required.

For future development of the energy efficient smart home we plan to optimize the performance of reading energy consumption. Currently reading one Plogg takes 1,2 seconds on average. When thinking of a fully equipped smart home, such a response time would be inacceptable.

We also would like to extend the means of communication between energy providers and the system and study more how this could benefit both sides.

Acknowledgment

This research was supported by the European Commission within the InterMedia Network of Excellence (project No. 038419).

References

[1] A. Al-akkad, F. Pramudianto, M. Jahn, and A. Zimmermann. Middleware for building pervasive systems. In *International Association for Development of the Information Society (IADIS): International Conference Applied Computing Rome*, pages 1–8, Rome, 2009.

[2] E. Bier, M. Stone, K. Pier, W. Buxton, and T. DeRose. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 73–80. ACM New York, NY, USA, 1993.

[3] A. Capone, M. Barros, H. Hrasnica, and S. Tompros. A New Architecture for Reduction of Energy Consumption of Home Appliances. In *TOWARDS eENVIRONMENT, European conference of the Czech Presidency of the Council of the EU*, 2009.

[4] S. Darby. The effectiveness of feedback on energy consumption. *A Review for DEFRA of the Literature on Metering, Billing and direct Displays*, (April), 2006.

[5] H. Eikerling, G. Graefe, F. Roehr, and W. Schneider. Ambient Healthcare Systems - Using the Hydra Embedded Middleware for Implementing an Ambient Disease Management System. In *Proceedings of the Second International Conference on Health Informatics*, pages 82–89, Portugal, 2009.

[6] M. Eisenhauer, P. Rosengren, and P. Antolin. A Development Platform for Integrating Wireless Devices and Sensors into Ambient Intelligence Systems. *2009 6th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops*, 00(c):1–3, Juni 2009.

[7] T. C. G. (GEsI). SMART 2020: Enabling the low carbon economy in the information age., 2008.

[8] M. Jahn, F. Pramudianto, and A. Al-akkad. Hydra Middleware for Developing Pervasive Systems: A Case Study in the e-Health Domain. In *1st International Workshop on Distributed Computing in Ambient Environments (DiComAe 2009)*, pages 13–21, Paderborn, 2009.

[9] T. Kirkham, D. Savio, H. Smit, R. Harrison, R. Monfared, and P. Phaithoonbuathong. SOA middleware and automation: Services, applications and architectures. *2008 6th IEEE International Conference on Industrial Informatics*, pages 1419–1424, Juli 2008.

[10] F. Milagro, P. Antolin, P. Kool, P. Rosengren, and M. Ahlsén. SOAP tunnel through a P2P network of physical devices. In *Internet of Things Workshop, Sophia Antopolis, Sep*, 2008.

[11] J. Müller, M. Jentsch, C. Kray, and A. Krüger. Exploring factors that influence the combined use of mobile devices and public displays for pedestrian navigation. In *Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges*, pages 308–317. ACM New York, NY, USA, 2008.

[12] C. Prause. Hydra's open source activities. *Hydra Newsletter*, 4:3–4, 2009.

[13] R. Reiners and M. Jentsch. Discovery and Interaction in Smart Environments. In *Ambient Intelligence and Human Security*, pages 135–138. ACM, 2009.

[14] R. Reiners, A. Zimmermann, M. Jentsch, and Y. Zhang. Automizing home environments and supervising patients at home with the hydra middleware. *Proceedings of the first international workshop on Context-aware software technology and applications - CASTA '09*, page 9, 2009.

[15] J. Rekimoto. NaviCam- A magnifying glass approach to augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):399–412, 1997.

[16] C. Ressel, J. Ziegler, and E. Naroska. An approach towards personalized user interfaces for ambient intelligent home environments. In *2nd IET International Conference on Intelligent Environments, 2006. IE 06.*, 2006.

[17] M. Rohs, J. Schoening, A. Krueger, and B. Hecht. Towards real-time markerless tracking of magic lenses on paper maps. In *Adjunct Proceedings of the 5th Intl. Conference on Pervasive Computing (Pervasive), Late Breaking Results*, pages 69–72. Citeseer, 2007.

[18] D. Wagner and D. Schmalstieg. First steps towards handheld augmented reality. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, page 127. Citeseer, 2003.

[19] G. Wood. Dynamic energy-consumption indicators for domestic appliances: environment, behaviour and design. *Energy and Buildings*, 35(8):821–841, 2003.