might sound contradictory at first. But, why not doing the rehabilitation training on a real bicycle outside in the forest?

E. *Delay-Tolerant acquisition of training data*

A BAN enables us to record vital parameters and transmit them wirelessly, but instead of just "dumb" sending data to a home platform, data can be processed inside the BAN as well. Watching thresholds for vital parameters is not that resource-intensive that it could not be done on small sensor nodes and algorithms for detecting ventricular fibrillation are also not very computation intensive as well. Connecting a mobile phone to a BAN for setting up a call in case of an emergency is also not very complex. Nevertheless the case of an emergency is not the standard case: As mentioned above the aim is to constantly monitor vital parameters and store the data for further analyses. Thus, equipped with some memory and the ability to "synchronize" data after returning home again, the pure monitoring of vital parameters can also be done when a person is not at home.

In the environment for aging platform a Delay Tolerant Network (DTN) can used to transport monitoring data from the BAN of rehabilitation patients to a health care datacenter. The BAN connects to the users smart phone via Bluetooth or ZigBee. The monitoring data is stored on the smartphone or on some gateway node until the patient passes an access point, to be forwarded via the Internet to the a residential gateway, so that the progress of the rehabilitation program can be reviewed by medical personnel. These reviews are only conducted every few days, therefore the monitoring data is not time critical and it is advantageous to transport it via the DTN, which does not cause any transmission costs. Of course the smartphone triggers a real time alarm or an emergency call via a GSM connection if the BAN detects critical or even life-threatening situations. But for rehabilitation, these situations are very rare, so normally there is no need for an expensive and energy consuming GSM link. Furthermore, no action by the patient is required to initiate a data transfer from the BSN to the health datacenter.

A typical use case is a patient doing outdoor activities, such as walking or jogging, for rehabilitation. At home, the BAN is connected via the SOHO-router to the datacenter. But as soon as the patient leaves his home, this short range wireless connection is disrupted. Maintaining a constant GSM link via the smartphone drains its battery and causes high costs, which would significantly reduce the systems user acceptance. Using DTN, the data is stored on the smartphone. As soon as the patient comes home (or passes an access point) the data is transmitted via WLAN from the smartphone to the SOHO router, which in turn forwards the data via the Internet to the healthcare datacenter.

III. Environments for Aging Platform

In the previous sections there have already been several labels for the underlying "Environments for Aging"-platform – that is because of it's universal structure. It surely is a "middleware" running on a (somehow embedded) "home-platform" that acts as a "residential gateway". As now dealing with the technical aspects we are willing to provide a more universal name for it.

A. *MSHP – Multi-Services-Home-Platform*

The Multi-Service-Home-Platform (MSHP) is a generic technical platform for "Environments for Aging" or ambient assisted living (AAL) and is based on the OSGi-Service-Platform. It is developed and used in the project and provides the basic infrastructure for a complex healthcare and monitoring system for elderly people at their homes. In the current stage of development and implementation the MSHP consists of a small number of standard PCs with many different types of sensors attached. It acts as a residential gateway with many different roles and services. The described use cases in section II differ in some needed services and attached sensors, but basic functionality and fundamental services are needed for all of them. Figure 1 shows the overall architecture of the MSHP.

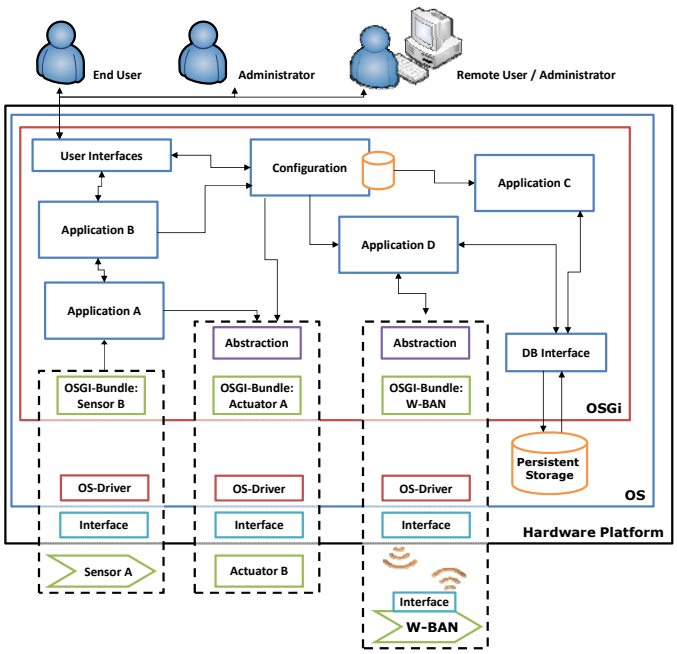

Figure 1. MSHP-Architecture

The hardware platform used within the project is currently a standard PC running a Linux (Ubuntu Studio). Besides providing hardware drivers for the attached devices there are only two major tasks for the operating system: A Java virtual machine as basis for the OSGi Service Platform and a SQL-database for the persistent data storage. Most of the functionality is realized by so-called bundles within the OSGi Service Platform, but there are also some more "intelligent" sensors attached as well, that relief the system from heavy computation jobs (image- and audio processing) at this stage. Nevertheless it is planned to integrate these modules as well inside the MSHP in the future.

For every sensor or actuator physically attached, an OSGi bundle representation inside the OSGi service platform is present. Bundles can be updated and exchanged during system runtime and OSGi takes care of dependency solving between different bundles and versions. Reusability and modularity is increased by an abstraction layer: For an application that monitors e.g. an ergometer for rehabilitation sports it is irrelevant and unpleasant do deal with vendor specific details.

978-1-4244-6672-6/10/$26.00 ©2010 IEEE

The abstraction layer just delivers standardized value sets to the monitoring application – regardless which manufacturer build the device.

B. Infrastructure Services

Within the MSHP there are some infrastructure services which are used by many bundles and relate to most of the use cases. Some of them are presented here:

1) Data Fusion: Localization

Localization is just one example for data fusion: localization information can be gathered by various sensors. Light barriers and door switches can be used to determine whether a room is entered or left, infrared- and ultrasonic sensors are able do detect the presence of a person in a certain room; cameras can even specify the approximate position of a person. Additionally light switches and powering on certain devices can predict the actual whereabouts of an inhabitant. All these data is assessed in a weighted map that is used to determine the position of a person with a certain degree of probability. Applications like e.g. activity monitoring mentioned above then can use this information for further processing.

2) Alarm Routing

Several use cases require the possibility of setting up an emergency call in case of a detected urgent situation, but also messages with lower priority have to be handled: After a detected behavioral change or a risen risk to fall not the ambulance is to be called, but certain specialists or relatives have to be informed. The alarm routing takes care of all that: Different roles and priorities can be defined for different applications. Even alarm chains with feedback channels can be used to ensure that there is an appropriate reaction within a given time [5]. Physical communication interfaces for the alarm routing can be PSTN / ISDN and DSL, but also UMTS / GSM. Signaling applications can be "normal" telephone calls (voice-over-IP as well), fax, SMS and E-Mail.

3) Connection to existing health telematics.

Necessarily, existing health telematics systems can be connected via various interfaces. A direct connection using a leased line or a modem connection is as well possible as connections via the internet realized through virtual private networks (VPN) or secured web services.

4) Persistent data storage.

Many applications require a persistent storage of data. Be it to do further calculations on historic data for activity monitoring or for a physician's view on recorded ECG-diagrams. There are many different types of data to be stored, e.g. high-frequent ECG raw data or medium large audio samples and a flexible database is the most likely and common way to keep and handle such huge amounts of data. Performance issues and SQL-accessibility on OS level (for back-up and maintenance tasks) were the most important reasons not to integrate the database into OSGi, but rather to have it running at operating system level. That is why we developed two different interfaces bundles to access standard SQL databases from inside the OSGi. One just passes SQL-statements; the other provides another abstraction layer and handles objects. Within the project MySQL is used. A central back-up service takes care of transferring the accruing data as well as configuration data and user settings to a datacenter via an internet connecting, where it is stored in an encrypted way.

C. Attached Sensors and Devices

As mentioned in the section II and III, there are several external devices connected to the MSHP via various interfaces.

- I²C-bus connects ultrasonic sensors for localization.
- EIB, KNX or LON-bus interacts with home automation devices like reed contacts, IR-sensors, outlets, and normal switches.
- USB- and "line-in"-Microphones are used for user-interaction and localization.
- FireWire (IEEE 1394)-Cameras for fall-detection /prevention and localization.
- PowerLine and proprietary protocols and interfaces for connecting "intelligent" white goods, like cooker, fridge or oven.
- A Bluetooth (IEEE 802.15.1) interface is used to communicate with "worn" accelerometers.
- IEEE 802.15.4 connects a wireless Body-Area Network with sensors for measuring vital parameters like electrocardiogram (ECG), heart rate, oxygen saturation, blood pressure, breathing rate and temperature.
- A telephone line (PTSN, ISDN or GSM) is needed for alarm routing.

For most of the "uncommon" interfaces mentioned above adapters to "regular" interfaces exist. EIB, KNX, or LON can be adapted to Ethernet (IEEE 802.3), WLAN (IEEE 802.11) or a serial line (RS-232); the same is true for the intelligent white goods. As the camera's image processing is done on a separate PC actually, the MSHP does not need a FireWire interface and image data can be transmitted via Ethernet as well.

That is why Ethernet, WLAN, USB, RS-232, I²C, IEEE 802.15.4 and IEEE 802.15.1 are the actual used physical interfaces for sensor connection and these interfaces will be needed on any possible integration platform.

IV. INTEGRATION ON A SOHO-ROUTER

Many of the services and use cases mentioned above require an established internet-connection and/or a telephone-line (PSTN or GSM/G3), which leads to the suggestion of combining these with standard broadband-access-routers and/or WLAN-access-points. There have been successful efforts in porting the OSGi service platform to embedded systems to assemble an integrated set-top box [6]. We are proposing the other direction to not develop another box but rather use an existing and well known basis.

A. Software Integration

As mentioned above, in the GAL-project currently a standard-PC equipped with Linux is used. Remarkable for software integration is that most common broadband access routers already are running an embedded Linux, which is

978-1-4244-6672-6/10/$26.00 ©2010 IEEE

normally transparent to the user, because of only being accessible through a web-front-end or some kind of proprietary configuration software. In earlier times many router-manufacturers did not invest much time in documenting their changes to the Linux-kernel, what led to the foundation of the OpenWrt-project [7]. This Open-Source-project maintains a model- and manufacturer-independent Linux-distribution, a package management system and a build environment [8]. Nowadays there are more than ten different hardware-architectures, more than 25 supported hardware-platforms (including normal PCs) and nearly a hundred common router-devices supported. This is why we have chosen OpenWrt as the basis of our integration.

1) Network Connections

As OpenWrt was intentionally developed as an operating system for routers, the connection to the Internet surely is one of the easiest parts. Several protocols including PPP and PPPoE, which are mostly used for common xDSL internet access, are supported. And as basic router functionalities there is support for Ethernet and various WLAN standards.

Having a telephone line and/or a GSM connection for emergency calls is essential. With porting the Asterisk [9] project to OpenWrt, the router becomes a complete telephone system by the way; including support for various connections like PTSN, ISDN, GSM and UMTS/3G. The telephone connection of any kind whatsoever can then additionally be used as a back-up Internet connection as well.

2) Java and SQL

For running OSGi the availability of a Java Virtual Machine is crucial. Luckily various Java implementations exist for embedded systems and there are installable packages for OpenWrt. Thus, installing OSGi is not a big deal any more. The persistent data storage requires a SQL database, which is the second critical part of the software integration. There are several SQL databases available for OpenWrt: Besides SQLite and PostgreSQL there is also MySQL, which we decided use within the project.

3) Delay Tolerant Network

There is current and ongoing work in developing needed features for OpenWrt [10]. Delay Tolerant Networking (DTN) is needed for the non-permanent link to the Wireless-Body-Area-Network. A Delay Tolerant Network is designed to be resistant against disconnections, network partitioning, extremely high delays and any other kind of disruption resulting in a non-existent end-to-end path between the sender and the receiver of a message. These messages are forwarded by routers just like on the "regular" Internet, but with the difference that each involved system (so called "DTN nodes") maintains a persistent long term message storage. DTN protocol data units, referred to as "bundles" (not to confuse with OSGi bundles in other parts of this document), are pushed into the storage whenever a link to the next hop is disconnected. Whenever a link is available the DTN bundles are pulled from the storage and transmitted to the next hop.

In a typical scenario DTN nodes are mobile devices equipped with short to mid range wireless interfaces such as WLAN, Bluetooth or IEEE 802.15.4 (e.g. ZigBee). These technologies use license-exempt spectrum and in contrast to GSM and UMTS do neither require an operator nor cause any charges to the user. Although these technologies cannot achieve long ranges they have significant advantages over cellular networks besides the fact that they are generally free of charge. Bluetooth and Zigbee offer a much better energy efficiency and WLAN reaches a many times better throughput. To overcome the range limitations DTN nodes use a store-carry-forward approach. Even in a very sparse network the nodes' movement almost inevitably leads to two nodes getting into each others range at some point of time. Therefore, the nodes are able to exchange DTN bundles. Later these nodes meet other nodes and exchange bundles again, and so on until the bundles reach their destinations.

DTN is standardized in RFC 5050 [9], specifying the Bundle Protocol and in RFC 4838 [12], which defines the DTN architecture. Several DTN implementations are available for various operating systems and platforms [10]. However, currently only IBR-DTN [13] is optimized for embedded systems such as SOHO routers, wireless sensor nodes and smartphones. IBR-DTN also integrates well into the software package management system of most embedded Linux Distributions.

B. Hardware Integration

When developing on and for OpenWrt there is no need in choosing a specific target-platform, as everything is easily portable. But, for the environments for aging platform, there is a demand for a number of physical interfaces. Besides that there is a higher demand on memory (main memory and storage) and processing power than there is present on very cheap home routers. We have chosen Texas Instruments' AR7-platform as a representative platform, because of being well known and widely spread. For demonstration purposes the "Ubiquiti RouterStation Pro" [14] is used due to many benefits:

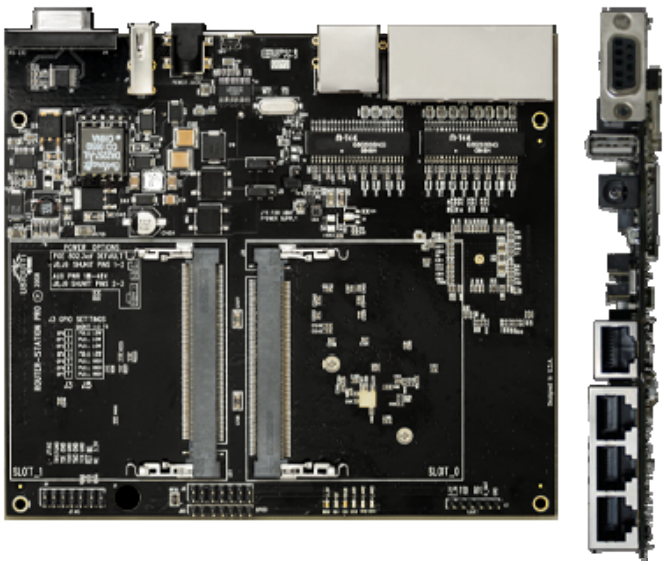

Figure 2. Ubiquiti RouterStation Pro (top-view and front-view)

There are already nearly all needed physical interfaces integrated: Ethernet (4 Ports: One for WAN-access, the others for LAN), WLAN (through a mini-PCI-card), USB, RS-232.

978-1-4244-6672-6/10/$26.00 ©2010 IEEE

Furthermore it is quite fast (Atheros AR7161 MIPS 24K @ 680MHz), fairly cheap (79 US$) and for data-storage a SD-card can be inserted.

The MSHP needs some more and different interfaces, but as mentioned before, most of them can be adapted to USB or RS-232. For IEEE 802.15.4 (for the W-BAN) and GSM (emergency back-up) there are existing and tested USB variants and a Bluetooth stick works as well.

V. Discussion and Conclusion

From a software-view, all basic parts of the MSHP are easily integrated in OpenWrt. There are several Java implementations that are able to run OSGi, but unfortunately there are differences in the supported Java versions – not regarding OSGi itself, but regarding some of the OSGi bundles that have been developed within the GAL project: As there were no precise instructions to use a minor version (e.g. Java 1.3) in the project, some bundles will have to be re-implemented to work on a SOHO-router as the available Java virtual machines do not compulsory have the complete and newest instruction set.

Within the GAL project by now there are some heavy jobs like image recognition or real-time audio processing done on separate and powerful machines. To get this functionality integrated into a SOHO router either specialized hardware has to be developed (e.g. network cameras with integrated high level image recognition) or using cameras for fall detection and prevention has to be given up.

As nearly all needed physical interfaces exist on (better equipped) SOHO-routers, only some work has to be done to integrate special hardware like the body area network. Nevertheless at the present time existing routers will not become an "environments for aging"-platform incidentally just by loading a new firmware. There are too many external devices needed for the presented use cases and nowadays a team of experts is needed to install the whole stuff.

Furthermore many of the sensors and actuators used in the GAL project require a wired connection by now, which in most cases can be adapted to present interfaces like USB. But having such a huge amount of wired connections is also a point of criticism as the wires must find their way through the habitation of an elderly person and loose cables on the floor undermine all efforts in fall prevention. Installing cables flush-mounted is much cost and work intensive for existing houses and usually only suitable for newly constructed buildings. Thus, for the future we will have to concentrate on equipping more and more sensors and actuators with wireless interfaces to decrease the need for wired connections.

Nevertheless we have shown that it is possible to integrate an "environments for aging" platform in SOHO-routers and there are many reasons to do so.

- Having fewer devices may reduce costs and error-proneness.
- Things that belong together (e.g. Internet routing and alarm routing) are integrated within one platform.
- An open source operating system and well known platform as basis makes it easier for future developments to be integrated.
- Current areas of research (e.g. DTN) can easily be carried to the end user.
- As we are using IPv6-adressing for the wireless sensor-nodes of the BAN, the customers' Internet infrastructure gets IPv6-ready casually.

References

[1] http://www.altersgerechte-lebenswelten.de/

[2] R. Haux et al., "Niedersächsischer Forschungsverbund Gestaltung altersgerechter Lebenswelten (GAL) - Informations- und Kommunikationstechnik zur Gewinnung und Aufrechterhaltung von Lebensqualität,Gesundheit und Selbstbestimmung in der zweiten Lebenshälfte - Zielsetzung und Arbeitsprogramm," in Zeitschrift für Gerontologie und Geriatrie, Vol. 41, Supplement 1, S. I/19, Steinkopff Verlag, 2008 (Published in German).

[3] A. Hein, M. Eichelberg, O. Nee, A. Schulz, A. Helmer, and M. Lipprandt, A Service Oriented Platform for Health Services and Ambient Assisted Living, Proceedings AINA Workshops/Symposia 2009, IEEE 23rd International Conference on Advanced Information Networking and Applications, Bradford, United Kingdom, 26-29 May 2009, pp. 531-537.

[4] E.M. Meyer, W. Heuten, M. Meis and S. Boll, Multimodal Presentation of Ambient Reminders for Older Adults, Proceedings Ambient Assisted Living 2010, Berlin.

[5] A. Helmer, M. Eichelberg, M. Meis, M. Gietzelt et al., Ein System zur eskalierenden Notruf- und Informationsweiterleitung vom Heimumfeld älterer Menschen zu externen Anwendern [A System for escalating emergency and information forwarding from the home of elderly people to external users]. Proceedings Ambient Assisted Living 2010, Berlin.

[6] T. Frenken, M. Lipprandt, R. Kluthe, and A. Hein, Eine kosteneffektive und flexible AAL-Plattform am Beispiel eines integrierten Versorgungs-prozesses [A Cost-effective and Flexible Technical Platform for Integration of Domestic Environments into Health Care Networks]. Proceedings Ambient Assisted Living 2010, Berlin.

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

[8] F. Fainelli, "The OpenWrt embedded development framework, " FOSEDM '08, Brussels, Belgium, 2009.

[9] http://www.asterisk.org/

[10] M. Doering, S. Lahde, J. Morgenroth and L. Wolf, "IBR-DTN: An Efficient Implementation for Embedded Systems", in Proceedings of the Third Workshop on Challenged Networks, CHANTS 2008, San Francisco, California, USA, September 15, 2008, pp. 117–120, September 2008.

[11] K. Scott, and S. Burleigh, Bundle Protocol Specification, IETF RFC 5050, experimental, November 2007.

[12] V. Cerf *et al.*, Delay-Tolerant Network Architecture, IETF RFC 4838, informational, April 2007.

[13] http://www.ibr.cs.tu-bs.de/projects/ibr-dtn/

[14] http://www.ubnt.com/products/rspro.php.

978-1-4244-6672-6/10/$26.00 ©2010 IEEE

Home

978-1-4244-6672-6/10/$26.00 ©2010 IEEE