Living++: A Platform for Assisted Living Applications

Muhammad Umer Iqbal
Networked Embedded Systems
University of Duisburg-Essen
Germany
umer.iqbal@uni-due.de

Marcus Handte
Networked Embedded Systems
University of Duisburg-Essen
Germany
marcus.handte@uni-due.de

Ngewi Fet
Networked Embedded Systems
University of Duisburg-Essen
Germany
ngewi.fet@uni-due.de

Pedro José Marrón
Networked Embedded Systems
University of Duisburg-Essen
Germany
pjmarron@uni-due.de

Stephan Wagner
Networked Embedded Systems
University of Duisburg-Essen
Germany
stephan.j.wagner@uni-due.de

UbiComp'13 Adjunct, September 8–12, 2013, Zurich, Switzerland.

ACM 978-1-4503-2215-7/13/09...$15.00.

http://dx.doi.org/10.1145/2494091.2497327

Abstract
Assisted living systems aim at increasing the independence of persons facing challenges in performing their daily tasks either due to cognitive diseases or aging. In addition, many systems try to help their caregivers in providing effective care to them. However, in order to be useful assisted living systems must be easy to use for the affected persons as well as for their caregivers. Furthermore, to be cost-effective the systems must be inexpensive. In this paper, we discuss our experiences from our assisted living system WebDA and based on the lessons learned we present the Living++ platform which strives to fulfill the mentioned goals. The Living++ platform is built on top of low cost and widely available devices, it provides a familiar TV-based user interface to its users and enables remote monitoring and management through freely available on-line services.

Author Keywords
Television, Assisted Living, Android, Framework

ACM Classification Keywords
D.2.11 [Software]: Software Architectures.

Introduction
Today, many societies are witnessing a significant demographic change. Fueled by a combination of factors

such as an increased life expectancy and a decrease in birth rates, they continue to age at a dramatic pace. In 2006, the European Commission [2] noted that the dependency ratio (the number of people aged 65 years and above relative to those aged from 15 to 64) will double and reach 51% by 2050. Consequently, this poses different challenges in providing effective health-care to the elderly persons either living alone or admitted in nursing homes.

From a technical point of view, one possible approach to mitigate this problem are assisted living systems and applications that empower the elderly persons to live independently for an extended amount of time. At the same time, the increased freedom provided by assisted living systems and applications can positively impact the quality of the elderly's life. However, in order to realize these benefits, ambient assisted living systems must be *easy to use* by the elderly persons. Furthermore, they must *integrate with the routines* of their care givers - since they are typically the ones who are administering the systems. Last but not least, the systems must be *extremely low cost* in terms of acquisition as well as operation.

In this paper, we discuss our experiences from our assisted living system WebDA [8] and based on the lessons learned, we present the Living++ platform. To be usable by a broad range of persons, Living++ provides a familiar TV interface that taps into *the ubiquitous display* found in virtually every home. To integrate with the routines of the care givers, Living++ uses freely available and widely used on-line services for both, remote configuration and monitoring. Finally, to exhibit very low acquisition and operation costs, Living++ is built on top of low cost Android hardware and is open to third party developments.

Next, we briefly discuss the WebDA system and experiences learned from its development and deployment. After that we present the high-level architecture of the Living++ platform followed by a section on some applications that we built with it. We then briefly outline related work and conclude the paper.

WebDA System

The WebDA system provides assisted living services to people with a beginning dementia. The primary goals of the WebDA system are helping people to localize misplaced objects, scheduling and reminding of appointments and training of communication and memory skills. Technically, WebDA relies on a traditional web-based architecture in which the user interfaces are generated as HTML pages by an application server that is installed on a PC in the elderly person's home. Internally, WebDA uses the OSGi framework to modularize the system and to support the addition of new modules over time. The elderly person as well as the care givers can access the applications using any browser but for our deployments we used IPADs.

Experiences

We deployed the WebDA system in homes of several elderly persons with varying degrees of dementia and performed technical tests as well as user studies. During our tests, we found out that the following system limitations can be mitigated with a different system architecture.

- *Higher hardware cost:* For WebDA, a typical installation encompasses a PC, a tablet and a WLAN router. Although, these systems may be available already, we found that most of our test persons did not have them.

- *Difficult installation:* For WebDA, it is necessary to install a PC. Depending on the model, size and heat generation of the PC, this can become challenging, especially, when considering smaller apartments. Due to the noise produced by a low-end PC and the fact that it must be running at all times, its placement in frequently used rooms can be highly undesirable for some persons and they may simply turn it off, thereby, deactivating the assisted living system altogether.
- *Arduous remote access:* As WebDA is based on web-based architecture, enabling remote access requires a certain amount of configuration. For example, to make the web server on the PC accessible, it is necessary to configure port-forwarding on the router. Furthermore, in order to enable convenient access, it is necessary to setup DNS. Finally, to enable secure communication, it is necessary to setup SSL certificates.
- *Difficult maintenance:* The maintenance of WebDA requires the installation or replacement of OSGi bundles. While it is possible to enable this remotely, e.g. by running an SSH server or configuring some remote desktop protocol, the steps required to do this often exceed the comfort level of care givers.

Based on our experience of developing and deploying the WebDA system, we specifically designed the architecture of Living++ platform such that it overcomes these issues. In the following, we describe the results in detail.

Architecture and Rationale

From a high-level point of view, the Living++ architecture follows the same overall architecture as most other assisted living systems. Devices installed in the home environment collect information by means of sensors and provide services to the elderly persons. To assist the care givers, (parts of) the information is made available through the Internet. Yet, when looking at the details depicted in Figure 1, Living++ exhibits noteworthy differences:

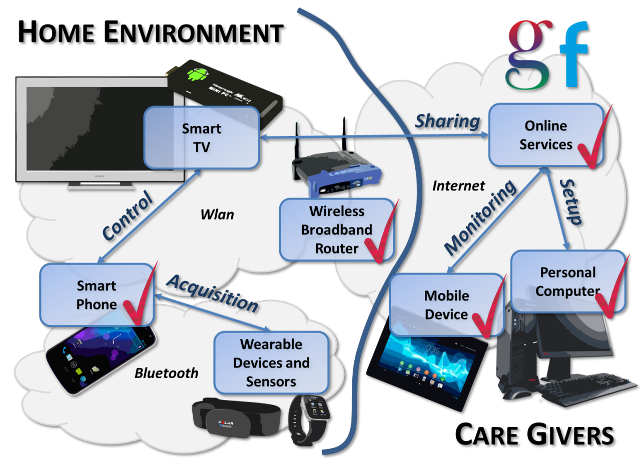

Figure 1: High-Level Architecture

- *Android as development platform:* In contrast to WebDA, the Living++ platform is built on top of Android. The reason for this is threefold. First, due to the availability of Android as open source, there are many low cost devices readily available on the market. Second, due to its heritage, it provides robust synchronization with multiple widely used on-line services such as Google Contacts or Calendar. Last but not least, using Google Play, it is possible to easily distribute applications and to keep them up-to-date without manual intervention. Together, these three factors make Android a good choice, since each of them contributes to reducing the cost.

- *Smart TV as service gateway:* To provide a simple user interface to elderly persons, the Living++ platform makes use of the TV as the single display that can be found in almost every home. In order to equip the TV with the necessary computational and communication capabilities, we rely on a Rikomagic MK802 II stick. The reason for this decision is its affordable price (below EUR120), its energy-efficient fan-less and thus, quiet operation as well as its hardware capabilities. The device is equipped with a 32-bit ARM processor and supports IEEE802.11. Using its HDMI port, the device can be plugged directly into the TV and it provides a micro SD card slot which enables the storage of data.
- *Smart phone for sensing and control:* To control the TV, the Living++ platform relies on a smart phone that connects to the Rikomagic stick using IEEE802.11 and provides the usual L-R-U-D-OK-BACK navigation that most TV sets are using as well. Besides controlling the TV, the smart phone is also used as primary sensing platform to enable activity monitoring. To do this, it can either use the sensors that are already built into the phone or it can integrate with other wearable devices and sensors via short range communication technologies such as Bluetooth.
- *On-line services for sharing and management:* To enable remote monitoring, the Living++ platform integrates with freely available and widely used on-line services. By exporting the sensor information to these services, the care givers can easily access them through various devices. Furthermore, by configuring the access policies of these services varying privacy goals can be realized. In addition, the Living++ platform also uses on-line services to enable remote configuration. By importing data from the services, a Living++ installation can be configured anywhere at any time through a robust user interface that is already familiar to many care givers.

Living++ Platform

Based on this system architecture, we first provide an overview of the components of the Living+++ platform. Thereafter, we discuss how the individual components interact with each other at runtime.

Components

The main components of the Living++ platform are depicted in Figure 2. Overall, the system consists of sensors and actuators, a smart phone doubling as remote control and sensing platform, an Android stick connected to the TV and on-line services accessible via the Internet. To be energy efficient and inexpensive, the sensors and actuators are typically equipped with custom firmware that cannot be manipulated easily. Consequently, their integration must be done by means of the remote interfaces provided by their manufacturers on the smart phone. A similar argument can be made for the on-line services. Just like with many sensors and actuators, the interfaces of the on-line services cannot be modified and thus, it is up to the smart TV to realize an appropriate integration. As a result, the main focus of the following discussion is the software for the phone and TV.

- *BASE middleware:* To enable communication between the smart phone and the smart TV, the Living++ platform uses our BASE middleware [4]. BASE provides support for spontaneous device interaction across different communication technologies and protocols. It supports dynamic device discovery and remote communication using different communication abstractions.

- *NARF component system:* To simplify the data acquisition using built-in as well as remote sensors, we use our NARF component system and its associated component toolkit [5]. With NARF, data acquisition applications are modeled as a set of software components with an associated wiring and parametrization. A component typically implements a single (reusable) function such as providing raw access to a sensor or performing a fast Fourier transformation on a signal.

- *Remote Control App:* The last remaining piece running on the smart phone is the remote control app. This app provides a simple input mechanism consisting of up, down, left, right navigation keys, an ok and a back button, a home button and a direct app launcher which is populated by the list of apps on the TV.

- *Event Logger:* In addition to BASE, the event logger is one of the main building blocks running on the smart TV. Its primary task is to manage a number of event logs that are populated by the data acquisition applications running on the smart phone. The logger maintains a set of local files that contain the events that have been transmitted to it.

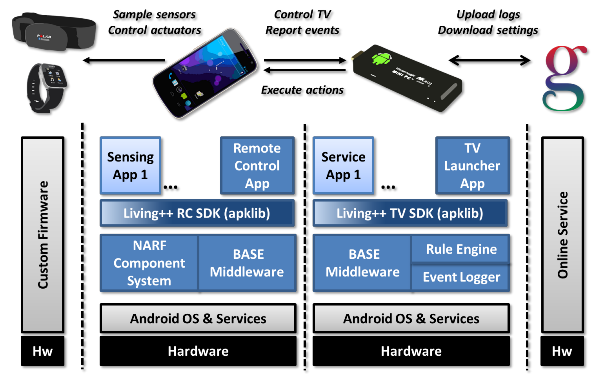

Figure 2: Living++ Components

- *Rule Engine:* Besides simply logging events, the smart TV is also able to use them to perform automation. To do this, it is equipped with a rule engine that can be configured with a set of event-condition-action rules that operate on the event logs managed by the logger. A rule can thereby be constructed as a boolean expression over multiple logs. This enables complex behaviors and allows the development of new apps on top of existing ones. To support different usage scenarios, the apps running on the smart TV can inject and modify the rules at runtime. When a rule fires, the associated action is executed. In order to keep the actions extensible, they are not directly associated with a piece of software. Instead, following Android's design philosophy, an action is simply an intent that is sent from the rule engine to another app which then takes care of handling them.

- *TV Launcher App:* The last remaining building block that is running on the smart TV is a launcher app that replaces the standard home screen. The

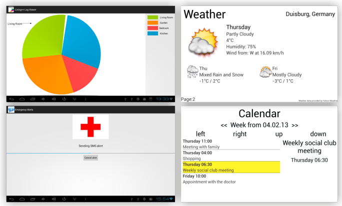

Figure 3: Living++ Applications (Log Viewer, Weather, Alert, Calendar)

launcher app primarily consists of a list view that can be controlled through the remote control app on the phone. In addition, it provides the remote control with a list of the installed apps which is used to populate a direct app launcher that is part of the remote control.

- *Living++ SDKs:* To develop further apps, the Living++ platform encompasses two SDKs, one for the smart phone and one for the smart TV. Both SDKs enable the interaction with the corresponding Living++ services through Android's RMI mechanism. The smart phone SDK enables the injection of configurations to perform data acquisition using the NARF component system. The smart TV SDK enables the injection of rules and the inspection of logs.

Interaction

To clarify the components described previously, we briefly outline their interaction at runtime. The interaction starts with the execution of apps on the TV and the smart phone. On the TV, the apps install rules into the rule engine. On the smart phone, the apps install configurations to acquire data from the phone's built-in or external sensors. When the components that are started by the configuration generate an event, the event is forwarded to the TV through BASE. When the event finally reaches the logger, it is written to a file and the rule engine is notified about the change. Optionally, the logged event can be written into a Google drive folder such that it can be accessed remotely by care givers. When the rule engine receives a notification, it evaluates the rules that are currently installed and that may be affected by the event. If the condition of a rule applies, the corresponding action is fired.

Applications

To test the Living++ platform, we have developed a number of assisted living applications on top of it. In the following, we briefly discuss some of them.

Applications

The applications that we have built with Living++ span a broad spectrum ranging from providing convenience functionality to health related monitoring. In the following, we briefly describe their functions and how they leverage different capabilities of the platform. Figure 3 and Figure 4 show some of these application.

- *Weather:* Weather is a simple TV app that is controlled through the remote control on the smart phone. Using left-right-ok and a keyboard that is displayed on the TV, a person can define places through which he can navigate using the up and down keys.

- *Calendar:* The calendar TV app displays the person's weekly agenda. The person can scroll through the different weeks using left-right buttons and scroll through different events using up-down buttons. Details can be displayed by pressing ok over a calendar item. The calendar app uses Google calendar as a backing data storage.

- *Localization:* The localization app is installed on the smart phone and provides indoor localization at room-level granularity with an update rate of approx. 0.5Hz. It relies on WLAN fingerprinting similar to RADAR and incorporates the NARF components used for LOCOSmotion [3].

- *Activity and Vital Sign Monitoring:* Similar, to indoor localization, these two apps consists of a set

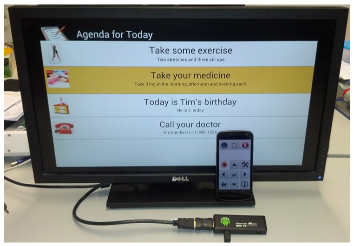

Figure 4: Living++ TV with Remote Control

of NARF components running on the smart phone which report their measurements to the TV such that they can be used in rules or exported to a Google drive. Activity monitoring taps into the accelerometer of the phone to determine whether the person is standing, sitting or walking. Vital sign monitoring integrates with a wearable pulse belt that reports its measurements to the smart phone through Bluetooth.

- *Log Viewer:* The log viewer app enables visual inspection of logged data produced by the localization and monitoring apps.
- *Context-aware Reminder:* Context-aware reminder is a TV app that can be used to configure rules that issue reminders depending on the context for example, it can be used to issue reminders only at certain places and at certain times.
- *Context-aware Alert:* Similar to reminders, context-aware alert is used to define rules over the logs generated by other apps. By integrating with the vital sign monitoring, for example, it is possible to generate emails or send SMS messages when the pulse exceeds a predefined threshold.
- *Fall Detection:* Fall detection can be seen as a special combination of activity monitoring and context-aware alert. However, instead of sending an asynchronous message, the TV app can establish a phone call using the smart phone, once a fall has been detected. To detect falls, we have implemented the approach described in [10].

Related Work

The instrumentation of home environments in order to assist their inhabitants has been the focus of a number of initiatives such as the Aware Home [6], Age-in-Place [1] or the Oatfield Estates Cluster [11] to name a few. While many of our applications are derived from such pioneering efforts, early research was leveraging hardware deployments that are comparatively costly. In contrast to this, with Living++ we are focusing on a hardware setup that can be easily deployed in many home environments.

From a functional and high-level architectural perspective, Living++ is similar to many other systems such as [12],[7] including our previous work on WebDA [8] in that it aims at providing assistance functions to both, the elderly persons as well as their care givers. However, instead of building on top of traditional PCs, which, based on our experiences, are often not available and cumbersome to install, Living++ relies on a low-cost Android stick that can be attached to *the ubiquitous display*, i.e. the TV.

Consequently, Living++ can be thought of a concrete implementation of the architecture described in [9]. However, as opposed to using a set-top box as sensor integration platform and a distributed database for remote monitoring, we use a smart phone that doubles as a remote control and freely available on-line services. Based on our experiences, it is usually not possible to enable reliable and apartment-wide communication with low-power communication technologies and the use of existing services can significantly reduce the development and maintenance cost of the system.

Conclusions

The Living++ platform aims at fulfilling the challenges of providing ambient assisted living applications to elderly people by relying on inexpensive hardware and robust Android system functions, by providing a familiar TV-based user interface to the elderly person and, by using freely available on-line services for remote monitoring and administration. Based on our experiences with previous systems, the resulting system is typically less expensive, easier to install and maintain while providing a similar level of service.

Acknowledgements

This work is partially supported by Ubicitec (European Center for Ubiquitous Technologies and Smart Cities), WebDA funded by the BMBF under contract number 16SV4023 and by LIVING++ funded by the BMWi under contract number KF2095019FR0.

References

[1] Dishman, E. Inventing wellness systems for aging in place. *Computer 37*, 5 (may 2004).

[2] European Commission. The demographic future of europe – from challenge to opportunity. In *Commission Communication, 571 final* (October 2006).

[3] Fet, N., Handte, M., Wagner, S., and Marron, P. J. Locosmotion: An acceleration-assisted person tracking system based on wireless lan. In *EvAAL Workshop, AAL Forum* (Eindhoven, Netherlands, Sept. 2012).

[4] Handte, M., Wagner, S., Schiele, G., Becker, C., and Marron, P. J. The base plug-in architecture - composable communication support for pervasive systems. In *7th ACM International Conference on Pervasive Services* (July 2010).

[5] Iqbal, M. U., Handte, M., Wagner, S., Apolinarski, W., and Marron, P. J. Enabling energy-efficient context recognition with configuration folding. In *International Conference on Pervasive Computing and Communications (PerCom)* (March 2012).

[6] Kidd, C. D., Orr, R., Abowd, G. D., Atkeson, C. G., Essa, I. A., MacIntyre, B., Mynatt, E. D., Starner, T., and Newstetter, W. The aware home: A living laboratory for ubiquitous computing research. In *2nd International Workshop on Cooperative Buildings, Integrating Information, Organization, and Architecture*, CoBuild '99 (London, UK, UK, 1999).

[7] Lopez, D., Blanco, S., Laiseca, X., and Diaz, I. Eldercare: An interactive tv-based ambient assisted living platform. In *Activity Recognition in Pervasive Intelligent Environments*, Atlantis Ambient and Pervasive Intelligence. 2011.

[8] Mohamad, Y., Gappa, H., Pullmann, J., Nordbrock, G., Velasco, C., Handte, M., Wagner, S., and Schweda, M. Context-aware support for people with dementia and their families. In *Deutscher AAL-Kongress 2012* (Berlin, Germany, Januar 2012).

[9] Sorwar, G., and Hasan, R. Smart-tv based integrated e-health monitoring system with agent technology. In *26th Int. Conf. on Advanced Information Networking and Applications Workshops* (March 2012).

[10] Sposaro, F., and Tyson, G. ifall: An android application for fall monitoring and response. In *International Conference on Engineering in Medicine and Biology Society (EMBC2009)* (September 2009).

[11] Stanford, V. Using pervasive computing to deliver elder care. *Pervasive Computing, IEEE 1*, 1 (2002).

[12] Wang, Q., Shin, W., Liu, X., Zeng, Z., Oh, C., AlShebli, B., Caccamo, M., Gunter, C., Gunter, E., Hou, J., Karahalios, K., and Sha, L. I-living: An open system architecture for assisted living. In *IEEE International Conference on Systems, Man and Cybernetics (SMC '06)*, vol. 5 (October 2006).