SmartAssist - Wireless Sensor Networks for Unobtrusive Health Monitoring

Andreas Schrader, Darren Carlson and Peter Rothenpieler
Institute of Telematics at the University of Lübeck

Abstract. Emerging demographic changes require new technical solutions to serve the growing elderly population in many countries around the world. Ambient Assisted Living approaches promise to deliver context-aware, personalized services designed to help elderly people live healthier and more autonomous lives. In this direction, we are developing the SmartAssist project, which provides a socio-technical platform based on wireless sensors at home and on the move that detect subtle long-term changes in health status and serve to enhance the user's social network with context-aware services. In this paper, we discuss the combination of wireless in-house sensors and mobile sensors on smart phones to realize unobtrusive health monitoring.

Keywords. Ambient Assisted Living, Sensor Networks, Ambient Health Monitoring

Introduction

The increasing age of our societies, combined with increasing demands for sustained, self-defined and autonomous at-home living, pose high demands on healthcare and service domains. Increasing medical costs and percentage of single households with higher risks of isolation are additional challenges. Consequently, supportive technical systems are seen as a major contribution for Ambient Assisted Living (AAL) with improved quality-of-life.

AAL systems combine recent technical developments like high-performance mobile processing, health sensors and advanced interaction technologies to realize interactive, adaptable and context-aware services that can be personalized to the respective customer. In order to achieve user acceptance for this potentially privacy-invasive approach, the main ingredients are unobtrusiveness with simple installation, reduced maintenance and transparent usage; cost efficiency with lightweight simple technical infrastructure; and flexibility with open interfaces for integration of third-party provided services.

In SmartAssist we focus on the social network of the user, comprising personal *patrons* (e.g., neighbors, friends, relatives, service staff, etc.) [7]. An in-home sensor network performs user activity monitoring. The algorithmically derived aggregated health status in combination with edited profile information of the user can be shared with patrons in order to realize context-aware services. Users can manage their own sensors, transparently observe the data collection and specify their access control and privacy settings. Patrons, which can be close relatives or medical doctors, can use the platform to monitor the condition of their seniors and communicate with them while external third-party providers can use the system to offer health or lifestyle services to the users.

1. System Architecture

1.1. Wireless Sensor Network

In order to support easy installation and unobtrusive operation, we do not apply any invasive sensor types, like cameras or microphones. Instead, a number of simple sensors for temperature, pressure, proximity, light, water and electricity are used. The sensors are embedded into very small form factor sensor nodes, which are able to automatically create an ad-hoc wireless network in the user's home. Our goal is to allow the healthcare staff to identify relevant spots in the house (e.g., coffee machine, bedroom door, bathroom water tap, etc.) and to attach the sensor boxes at the respective positions without any additional installation tasks. Battery life in our tests guarantees at least one year of permanent operation without any maintenance. A gateway device collects the sensor data and sends aggregated information to the SmartAssist server over the Internet or mobile phone network. Data aggregation is performed automatically using unsupervised self-learning signal processing algorithms for detecting relevant events out of the raw sensor data (e.g., coffee machine not used today), which could trigger social services (e.g., neighbors knocking at the door), professional services (e.g., cleaning or food delivery), or professional medical services (from healthcare visits to emergency response).

1.2. Online Platform

The SmartAssist online platform is the central access point for the user, the communication center for the social network members, and the integration point for third-party developers. The platform is implemented using the content-management-system Drupal 6 [1] with additional modules to support accessibility, manage access control and the creation of relationships between users (patron/senior). The integration of external service providers is realized using the Apache Shindig container[2], which is the open source reference implementation of the OpenSocial standard.

The OpenSocial container enables external providers to integrate their applications (gadgets) into the platform and to easily access the information stored within SmartAssist. Gadgets are XML files following the OpenSocial gadget specification, containing meta-data about the gadget as for instance the gadget author, description, features and dependencies as well as the gadget content itself which can be HTML, CSS, JavaScript and external service calls. Gadgets are written by third-party providers and are added to the SmartAssist gadget repository from where the users can add them to their profile. To display the gadget, it is transformed into HTML in real-time by the OpenSocial container and dynamically integrated into the user interface. The OpenSocial container provides different JavaScript Application Programming Interfaces (APIs) to the gadgets, which enable user interaction, data persistence,

[1] Drupal – Open Source CMS, http://drupal.org
[2] Apache Shindig, http://shindig.apache.org

authorization and access to the users' information. As described in [7], making the personal information and sensor data accessible to external service providers leads to a variety of privacy risks, which in SmartAssist are dealt with using mechanisms and technologies such as OAuth. Further details can be found in [8].

The OpenSocial container further offers a (RESTful) web service API for providing access to the data from a remote service consumer, which is not only utilized to give access to third-party providers but is also used to provide access to the users data through his own mobile device.

1.3. Mobile Components

To facilitate the rapid creation of mobile SmartAssist applications, we are also developing a context-aware middleware framework called Aladdin [3]. The Aladdin framework supports a broad range of context-aware applications through a lightweight, power-aware service model designed for deployment on resource-constrained mobile devices. During runtime, Aladdin continually analyses the user's environment using a set of dynamically installed context modeling plug-ins and securely provisions modeled contextual information to registered applications. Through the use of an extensible architecture and mobile code techniques, Aladdin is capable of modeling a broad range of commonly encountered context information without the need for widely deployed instrumentation or infrastructure. Related, we are also developing an Internet-based plug-in repository, whereby third-party developers can develop and integrate context plug-ins for use by the Aladdin community. Extracted contextual data are securely provided to SmartAssist applications, which can utilize the data to adapt and personalize their runtime behavior to help improve a user's overall experience. Context-aware runtime adaptation is beneficial in mobile assistive scenarios, where a user's physical state and attention may be limited [1].

In the Aladdin approach, domain-specific software running on a commodity device hosts the Aladdin framework (see Figure 1). Based on the Façade pattern [2], the framework provides a context management API and related set of context events. During runtime, Aladdin automatically analyses the capabilities of its host device and environment using dynamically installed capability analysis plug-ins. Based on the detected capabilities of the host device and environment, Aladdin then dynamically downloads and installs context acquisition and modeling plug-ins that are capable of providing high-fidelity native context data (NCD). During runtime, low-level context preprocessing and quantization are provided by the installed context plug-ins, resulting in the generation of context events (containing NCD) that are received by the hosting application. The hosting application may react to incoming context events as needed, according to their local application logic. Aladdin has been validated through the construction of three diverse application models, including a mobile interactive cinema platform [3]; a museum tour-guide system [4]; and a pervasive multiplayer tangible game [5]. Related work indicates that client-centric approaches, such as Aladdin, can be effectively adapted to large-scale heterogeneous environments [6].

Security and data privacy are important concerns for middleware frameworks involving contextual data, which often involve additional security considerations due to the sensitive nature of the modeled information (e.g., location or identity). To address

[3] Aladdin Framework, http://sourceforge.net/projects/aladdincontext/

these concerns, the Aladdin framework establishes extended security mechanisms to prevent unauthorized access to data and functions [7].

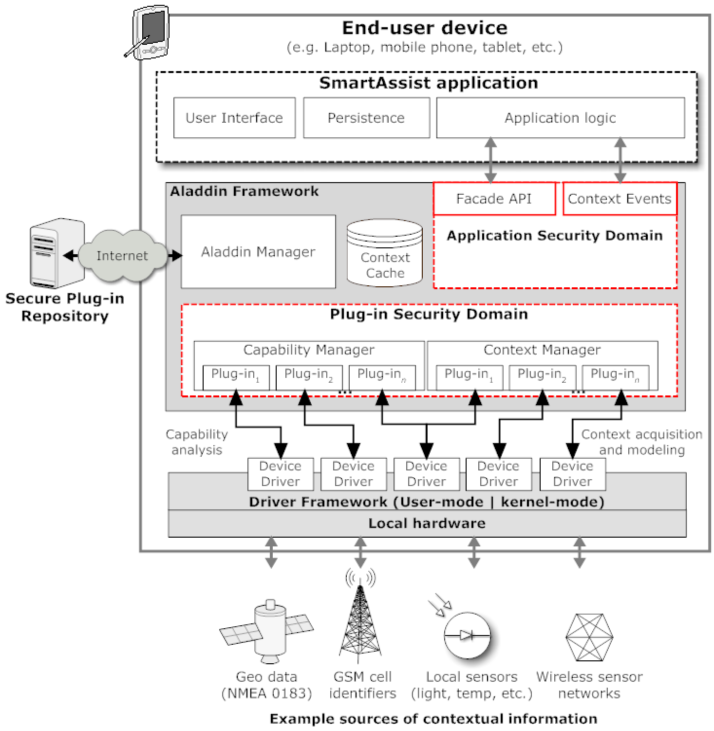

Figure 1: Overview of the Aladdin Framework architecture

2. Health Monitoring

The SmartAssist project does not focus on the usage of vital sensors to directly imply health status information. Instead, non-vital sensors in the house are algorithmically processed in order to indirectly derive long-term health status changes. In addition, sensors on the mobile phones might be used (e.g., position, accelerometer, light, etc.) in order to support the user on the move. Nevertheless, users might also want to include health-specific sensors for vital data like heart rate, glucose, etc. This can be easily supported by the provision of appropriate Aladdin plug-ins for communication with relevant devices, e.g., Bluetooth-based heart rate monitoring belts.

The following example service scenario is intended to outline the potential benefits gained from the combination of social data with stationary and mobile sensor values, not reachable in conventional AAL architectures.

2.1. The BikeWars Scenario

Anton and Bert are both SmartAssist users and each have a variety of sensors in their homes, monitoring, e.g., the temperature in their living-room and their current location.

Anton and Bert are old friends, but for several years have been living in different cities, making it difficult for them to undertake their healthy hobby together: bicycling. However, they both own computer controlled bicycle ergometers and are growing fans of the *BikeWars* gadget provided by their healthcare insurance and available on the SmartAssist platform. BikeWars allows them to challenge each other to bike races and to participate in tournaments using a combination of their home bicycle ergometers and their mobile devices (e.g., smart phone or tablet pc).

Using their mobile devices, Anton and Bert agree on one of the BikeWars racing tracks, e.g., a mountain climb in the Alps and start their race. BikeWars will then regulate the intensity of the exercise using combined context information consisting of profile information (age, fitness, and geographic preferences), values from the stationary in-house sensors (room temperature and humidity), data from sensors embedded in the mobile devices (accelerometer), information provided by the ergometer (pedal rotation speed, brake and gear settings, etc.), heart-rate belt values, and information from the exercise application (previously collected training data, contestants position on the chosen track).

Through the use of adaption mechanisms, the intensity can be normalized between both contestants, making the race more challenging for Anton, who is more physically fit, and more fun for Bert, who can still win against Anton, who is five years younger. The provider of the BikeWars gadget can then analyze the progress of its users, give advice for better training, display fitness or health and equipment related advertisements during the races (e.g., for a better training bicycle or accessories). In case of dangerous health parameters, the system could automatically inform the healthcare center, establish a voice communication channel for consultation or even place an emergency call. After a successful race, the data obtained during the training is injected to the personal health record and can be used to discuss and optimize the rehabilitation procedure at the next doctor appointment.

2.2. Technical Realization

BikeWars makes use of the gadget JavaScript API to display its user interface, access user information and to enable the exchange of messages between different users of the gadget. Since the OpenSocial API and the gadget specification both are used on a variety of different online platforms such as iGoogle, XING, all German VZ networks or MySpace, the applications based upon these standards are platform-independent and can be seamlessly integrated in any of the compatible containers. Anton and Bert are thus able to challenge each other via the gadget by sending a race-request to each other. This request is displayed within the SmartAssist platform and is automatically forwarded to their mobile devices in case they are not currently using the online platform.

The BikeWars application on the mobile phone is realized as an Android application using the Aladdin framework to communicate both with the SmartAssist server and with the used devices. Anton and Bert can connect their mobile devices to the ergometers using the Aladdin context plug-in provided by the bike vendor through the plug-in repository. Another plug-in provided by the open-source community is used to communication with the heart-rate belt devices and to receive measurement values. In order to access the temperature and humidity sensor data from the house, another plug-in provided by the SmartAssist framework will be used to access the server database in the background. Using the web service API, the same plug-in can inject the racing results and analyzed training data into the persistent application data store which is part of the user's server-side profile.

The dynamic download features of Aladdin would even allow for Anton to visit Bert and take a ride on his ergometer. Aladdin would automatically identify the new device type, request an appropriate plug-in from the repository and install it during run-time. This flexible scheme allows for dynamically and securely employing ubiquitous infrastructure without having to instrumentalize the environment.

3. Conclusions and Future Work

In this paper, we presented the SmartAssist architecture as a socio-technical system for supporting an autonomous life for elderly people. The SmartAssist approach combines a wireless network of stationary sensors with data gathering from sensors on mobile devices for realizing unobtrusive health monitoring at home and on the move.

The proposed example scenario of linked bike exercises outlines the potential strength of the system in terms of flexibility and scope for supported novel service scenarios. We are currently evaluating the system architecture by equipping about 50 single households of elderly people in the city of Luebeck, Germany, with a set of household sensors. The measurement results of a period of one year will be compared to a respective control group by counting rates of health-related events (e.g., falls, hospitalizations, etc.), performing quality-of-live assessments, and evaluating usability aspects with questionnaires.

The SmartAssist project is supported by the Federal Ministry of Education and Research (BMBF), Germany. ID: 16KT0942. For more information, please see http://www.smartassist.de.

References

[1] Ran, L., Helal, S. and Moore, S. 2004. *Drishti: An Integrated Indoor/Outdoor Blind Navigation System and Service*. In: Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04). IEEE Computer Society, Washington, DC, USA, April 2004, pp. 23-30.

[2] Gamma, E., Helm, R., Johnson, R. and Vlissides, J. M. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1994.

[3] Carlson, D., Schrader, A. and Busch, D. *Modular framework support for context-aware mobile cinema*. Personal and Ubiquitous Computing, 12, 4 (April 2008), 299-306.

[4] Carlson, D. *Aladin: An Extensible Ubiquitous Computing Infrastructure*. Master's thesis, University of Luebeck, Luebeck, Germany, 2005.

[5] Heumer, G., Carlson, D., Jung, B. and Andreas Schrader. *Paranoia Syndrome – A Pervasive Multiplayer Game*. In: Carsten Magerkurth, Carsten Röcker (eds.): Pervasive Gaming Applications - A Reader for Pervasive Gaming Research vol. 2, Shaker Verlag, pp. 179-198, November 2007.

[6] Riva, O. *Contory: A Middleware for the Provisioning of Context Information on Smart Phones*. In: Proceedings of Middleware 2006, LNCS 4290, Springer, pp. 219-239.

[7] Rothenpieler, P., Zwingenberg, H., Carlson, D., Schrader, A., and Fischer, S. *Datenschutz im AAL Service System SmartAssist.* 4. Deutscher AAL Kongress, Berlin, Germany, January, 2011. (to appear).

[8] Rothenpieler, P., Becker, C., and Fischer, S. *Privacy concerns in a remote monitoring and social networking platform for assisted living.* In: Sixth International PrimeLife/IFIP Summer School on "Privacy and Identity Management for Emerging Internet Applications throughout a Person's Lifetime", Helsingborg, Sweden, 2010.