Automizing Home Environments and Supervising Patients at Home with the Hydra Middleware

Application Scenarios using the Hydra Middleware for Embedded Systems

René Reiners, Andreas Zimmermann, Marc Jentsch and Yan Zhang

Fraunhofer FIT
Schloss Birlinghoven
53754 Sankt Augustin, Germany

{rene.reiners, andreas.zimmermann, marc.jentsch, yan.zhang}@fit.fraunhofer.de

ABSTRACT

In the Hydra Middleware project context awareness, among many other aspects plays an important role. Context is not only defined by the user's presence but also by identifying available devices and services that are offered by the environment or the devices themselves. Inside the architecture, special components handle contextual information by also separating between core-functionality and the possibility to extend the built-in concept by new components providing context information.

This work presents a first step towards ubiquitous and context aware applications in the healthcare and home automation sector. In the scope of the Hydra Middleware project, applications taken from different domains are derived from the current project's state and knowledge. The inclusion of different application domains supports the development of a domain-independent middleware and a wide spectrum of interests for application developers.

Categories and Subject Descriptors

K.4 [**Computers and Society**]: Public Policy Issues—*Computer-related health issues*; H.3 [**Information Storage and Retrieval**]: Systems and Software—*Distributed systems*; D.2 [**Software Engineering**]: Software Architectures

General Terms

Design, Experimentation, Standardization

Keywords

Context Awareness, Home Automation, Healthcare, Distributed Data Flow, Middleware

1. INTRODUCTION

Rapidly changing requirements and dynamic environments drive the development of context-aware applications. Research into context-aware computing focuses on programming frameworks and toolkits that support the development of context-aware applications ([1], [2], [3], [4]). The rapid prototyping of context-aware applications supplements an application of a user-centred design process, which yields to a richer understanding of the context of use and guarantees a certain degree of user acceptance.

Middleware-based approaches to context-aware computing try to hide low-level functionality for the acquisition, transformation, dissemination of context information. However, these approaches provide weak support of programming abstractions like the determination of situations or the access to historical context information. Furthermore, a context middleware mostly focuses on the acquisition of context information, and thus, insufficiently address control and actuation mechanisms of the targeted context-aware application.

The two case studies described in this paper document the application of the Hydra[1] middleware for the construction of context-aware applications, in order to prove the validity and general applicability of the software components and the software architecture behind. The Hydra middleware[2] aims at supporting application developers in the area of embedded systems through guiding the software engineering process and hiding complex underlying technical details. Hydra offers a software and device development kit, and compiles special-purpose software components as so-called Hydra managers in an architecture displayed by Figure 1. These Hydra managers relieve software developers from conducting recurring implementation tasks such as networking and discovering devices, securing communication, storing data, or retrieving information.

2. CONTEXT-AWARENESS IN HYDRA

The Hydra middleware introduces the term *Hydra-enabled device* in order to refer that its features are specified and the device is available in the Hydra network, i.e. an embedded system. The functionality of a hydra-enabled device is dis-

[1] *Heterogeneous Devices in a Distributed Architecture*
[2] http://www.hydramiddleware.eu

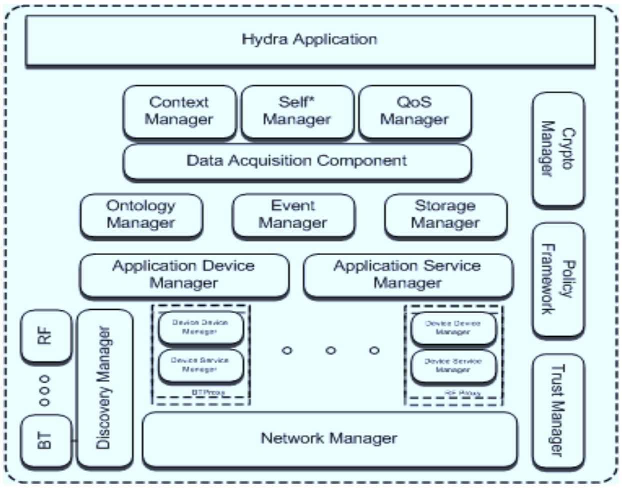

Figure 1: The Hydra Middleware Architecture.

tributed by offering web services. This way, semantics and descriptions can be provided via a WSDL file [3]. Extended components may ensure quality of service (Qos), e.g. in multimedia environments or check whether offered services match at what rate.

Hydra provides several procedures to network different types of devices. Resource-full devices, on which a core set of Hydra managers can be deployed and run on such a device, can immediately access the Hydra network. Resource-constraint devices such as sensors will interact with the rest of the Hydra network through a Proxy running on a dedicated gateway. Such proxies handle the communication with the resource-constraint devices and manage the protocol conversion to achieve an IP communication.

The Hydra middleware needs to offer an efficient way to share resources among the Hydra Network, in a scalable, distributed and efficient way. It also needs to prevent system failures when a node is not available, and allow ubiquitous access to the network. Therefore, the Network Manager implements JXTA as the Peer-to-Peer model for device to device communication.

On the right hand side of figure 1, the Crypto, Trust and Policy Managers take care for cryptographic operations, the evaluation of trust in different tokens and the enforcement of access control security policies. Hydra introduces a layered architecture of context-aware applications:

- The Context Data Acquisition Layer manages the sensor devices, senses context information and performs plausibility checks.

- The Context Management Layer manages the retrieval and the lifecycle of context.

- The Context Awareness Layer supports context-awareness based on a context knowledge base.

- The Context Interpretation Layer reasons on context, performs a semantic processing and performs context-sensitive operations.

[3]WSDL = Web Service Description Language

A specific set of Hydra managers realise this layered model: The Application Device and the Application Service Managers provide programming interfaces and information for the different sensor devices to the software developers. The Data Acquisition Component retrieves the data delivered by the sensors (via push or pull mode) and check the values for plausibility.

The Context Manager allows for the definition of an application-dependent context model using key-value pairs or OWL/SWRL ontologies managed by the Ontology Manager. The Ontology Manager is also involved in the reasoning about context information and the semantic processing[5].

Furthermore, the Event Manager provides a topic based publish-subscribe service for context information and the Storage Manager realises the persistent storage of this information in Hydra middleware.

3. APPLICATION SCENARIOS

As a starting point, different application domains were identified to conclude what requirements application developers would be confronted with when designing applications in these domains. The application domains were to be taken from very different fields in order to ensure the domain-independent applicability of the Hydra Middleware. The application domains also formulate different requirements concerning context awareness.

Home Automation: In this domain, context information concerning user presence and tracking, data collection from the devices in the environment and access control are in the main focus. Securing communication channels for data exchange or the transfer of access tokens for automatic door locks are important.

Healthcare: In the healthcare sector, medical devices, monitoring and also access rights to patients' home environments as well as confidentiality issues concerning patients' files play important roles. Also the management of access rights to certain details of the patient's files, for example, need to be considered.

3.1 Home Automation

For this application scenario, a self-monitoring and managing house which is capable of detecting errors and damages within the environment is introduced. So, damages on the heating system or a critical decrease of the temperature in several rooms caused by open windows can be detected. Based upon the owner's presence, different measures are taken to handle this situation. In the case that the house owner is present, he is directly informed via screens or speakers in his environment.

If he is far away, e.g. on a business trip, he receives a short message on his mobile phone explaining the current situation at home. He then has the opportunity to assign a repair task to a handcraft company to fix the damage. In order to grant access to his flat, he receives a list of trusted companies. After choosing one of them, a token generated by the house control unit is digitally signed to the company which then gains access to the flat. During the whole process, the owner is constantly updated via his mobile phone.

The demonstration could convey the idea of the Hydra middleware and at the same time show an interactive event and decision flow based on different situation parameters. This was rewarded by the "Runner's Up Demonstration

Figure 2: The demo stand for the eHealth application at Cebit Hannover (March 2009).

Award" at the ICT Mobile Summit at Stockholm in 2008[4]. At the ICT Event Lyon[5] (Nov 2008), the demonstration managed to be elected among the top ten exhibitions among nearly 200.

3.2 Healthcare

The Hydra Middleware eHealth application was presented twice this year: At the GSMA Mobile World Congress in Barcelona[6] (Feb. 16-19, 2009) and CeBit[7] Hannover (March 3-8, 2009).

A conceptual healthcare assistance system for patients who needed home supervision was created with the help of the Hydra Middleware. There are already existing but closed systems on the market that are for example provided by Philips[8]. Another example application is currently presented by Apple[9].

The scenario describes how the Hydra Middleware can be used in a healthcare scenario where vital functions of patients are constantly monitored and remotely accessible by a doctor in a hospital.

Additionally, the system provides personalized medical schedules after which the patients should perform special tasks. So for example, continuous measurement of the blood pressure might be needed. In order to guide the patient through the schedule and remind him of his tasks, a mobile device is used. Direct conversation or text-based messages can also be exchanged between doctor and patient. Messages sent out by the system to inform the patient, e.g. that the healthcare service is going to arrive, are also shown on the mobile device. Example screens taken from the mobile application are shown in figure 3. For the application, an Apple iPhone 3G was used because of is display and interaction capabilities.

In case that the patient is not able to open the door personally, e.g. in case of an emergency or due to his current state, access tokens are also activated by the system so that the doctor, ambulance or the healthcare service gain access

[4]http://www.ict-mobilesummit.eu/2008
[5]http://ec.europa.eu/information_society/events/ict/2008
[6]http://www.mobileworldcongress.com/
[7]http://www.cebit.com
[8]http://www.medical.philips.com/main/
[9]http://www.apple.com/iphone/enterprise/doylestown.html?sr=hotnews.rss

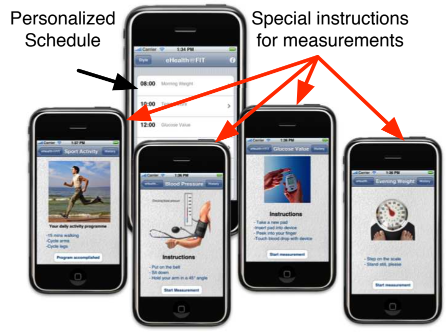

Figure 3: Screenshots from the mobile eHealth application.

to the patients' home. The token is stored on an RFID card which is presented at the automatic door lock at the patient's home. Access is only granted in case of an authorized visit or an emergency.

3.3 Devices and Infrastructure

For the realization of the home automation and healthcare scenarios, several different components were used in order to show the diversity of devices supported by the Hydra Middleware.

A *Sony Playstation 3*™ is used for storing the data which is collected by the medical devices, handles information about the owner's presence or decides whether a presented token at the door lock is valid. Token generation and message creation is also performed on this device. The connection to the network is established via LAN. Wifi would also be possible.

The *Apple iPhone 3G* is connected wirelessly to the network and runs the eHealth application for the patient. Screenshots of the application are summarized in figure 3. For the home automation scenario, it can also serve as standard phone displaying the system messages and being responsible for the digital token signature.

A *Nintendo's Wii Balance Board* serves as a Bluetooth weight scale. Since it is not possible to run web services on that kind of devices directly, the Hydra proxy approach (cf. chapter 2) is applied. Therefore, the Bluetooth connection is established with a notebook PC (proxy) that is capable of consuming corresponding web services on the Playstation 3 to set the currently measured weight value.

In order to measure several physiological values like heart rate, ECG, breath frequency, a *Becker Varioport* [10] sensor kit is integrated into the application. The sensor kit is integrated into the Hydra network via the notebook's serial port.

The door control is realized by an *RFID card reader* connected via USB to the Playstation 3, the door itself is controlled by a Lego™ motor controlled by an *Lego Mindstorms NXT™* brick[11]. For the home automation scenario, additional sensors are connected to the bricks in order to detect an increase in humidity at a heating system pipe or a sudden

[10]Medizinische Mess- und Informationstechnik
Dipl.-Ing.(FH) Klaus Becker,
Karl-Seckinger-Strasse 48, D-76229 Karlsruhe.
Tel 0721 / 94644-0 Fax 0721 / 94644-0
[11]http://mindstorms.com

decrease in temperature. Open doors an windows can also be detected. The connection between the NXT brick and the Playstation 3 is based on Bluetooth.

At the clinic, the doctor as well as the nursing service use a *tablet PC* which is connected via WiFi to the network. A smartcard reader supports the differentiation between the two roles: Doctor and nursing service. Depending on who is currently working with the terminal, the system reveals different details of information. The treating doctor retrieves the full patient's file whereas the nursing service only receives information currently necessary for the assigned task. This differentiation also shows the concerns about context awareness.

For the home automation scenario, the tablet PC is used in order to inform a service technician about a new assignment. Since he carries a writable RFID token with him, the system on his tablet PC supports the transmission of the token to this card.

4. FEEDBACK

The feedback collected during the different exhibitions of the applications scope was very positive. Many visitors gave the feedback that the aim of the application was very interesting and possesses a lot of potential for future extensions.

Context awareness of the application and the monitoring of the patient's state was not seen as a critical mechanism but a very important support. The collection of context information takes place in a non-obstrusive way such that also medical personal and visitors who were familiar with home supervision scenarios accepted the data collection activities.

Throughout the Hydra project, the concept of context awareness is more extended in the direction of device and service context, i.e. discovery, check of availability as well as interoperability. This allows the design of future application scenarios embedded in environment and therefore making them smarter.

Since the Hydra Middleware aims for supporting embedded systems and allows a high degree of freedom for application designers, the project group will continue research on integrating more semantic and context aware functionalities. Future work will cover the extension of the existing demonstrator and integrate more medical devices. One main reason for this is to evaluate the Hydra platform while it is under development.

Device and services ontologies as well as managers ensuring the secure and fail-safe connection between the different components of an embedded system will play an important role in the future progress of the project.

5. SUMMARY

The Hydra middleware addresses a lot of features as can be derived from chapter 2. The major issues that are covered by this work are summarized again in the following:

- Context awareness is provided in the scope of authorization, restrictions on information access, door access control and the supervision of he patient or the owner's presence state, respectively. If a situation occurs in which certain demands are not met, the system recognizes the erroneous state and reacts to it in a situation-specific way.

- Many kinds of connections, including serial ports, LAN, WiFi or Bluetooth, can be used within a Hydra environment.

- The Hydra Middleware is designed to run on resource restricted devices like Lego NXT bricks ™, the Wii Board or even the PS3™(regarding memory).

- The system runs within a distributed architecture, i.e. each device offers services (or data) that can be consumed by any other device.

- The whole inter-device communication can all be secured. The decision is made by the developer. The Hydra Middleware handles the technical realization.

6. REFERENCES

[1] H. Chen. *An Intelligent Broker Architecture for Pervasive Context-Aware Systems*. Ph.d., University of Maryland, 2004.

[2] A. Dey, D. Salber, and G. Abowd. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction, Anchor article for Special Issue on Context-Awareness*, 16(2-4):97–166, 2001.

[3] C. Efstratiou. *Coordinated Adaptation for Adaptive Context-Aware Applications*. Ph.d., Lancaster University, 2004.

[4] K. Henricksen and J. Indulska. Developing context-aware pervasive computing applications: Models and approach. *Pervasive and Mobile Computing*, 2(1):37–64, 2006.

[5] W. Zhang and K. M. Hansen. Semantic web based self-management for a pervasive service middleware. In *SASO '08: Proceedings of the 2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, pages 245–254, Washington, DC, USA, 2008. IEEE Computer Society.