AmICom - Middleware Support for Ambient Communication

Jan Koch* and Ingmar Fliege†

*Robotics Research Lab

†Networked Systems Group

Department of Computer Science, University of Kaiserslautern
Gottlieb-Daimler-Straße, 67663 Kaiserslautern, Germany

***Abstract*— Intelligent environments are based on services that are deployed on hardware nodes that communicate among each other. Generic and light-weight communication middleware for service interaction is the basis of the ubiquitous idea. Actuator, sensor and information systems as well multimedia devices are distributed over a set of hardware nodes. If service discovery, interaction paradigms, availability check, capability, and context description are already considered on the communication level, it is easier for service developers to implement robust, powerful and efficient applications. Learning from conventional embedded systems helps to understand the ambient view beginning at resource-constrained hardware and pragmatic communication paradigms. Naturally, embedded systems are already pervasive and ambient regarding service distribution, thus hiding the technology from the user. Interfaces are often only indirectly specified and the collaboration structure is static. In this work, AmICom, a new light-weight communication middleware for dynamic ambient applications is presented. This middleware is already in use on several nodes within a prototypical assisted living environment.**

***Index Terms*— Ambient Intelligence, Middleware, Assisted Living, Communication, Mobile Robotics**

I. Introduction

Communication is the key to intelligence. This paradigm does not only apply to humans but also to technical systems. If the vision of ambient intelligence as it is aspired by the community since the very beginning [1], one has to establish a communication system that fulfils a great number of requirements [2]. In this work we want to describe the state of our own approach to create a truly ubiquitous communication system. The communication middleware *AmICom* has been developed in the Networked Systems Group[1]. We will present the functionality of the AmICom middleware and the interoperation with several sensors and a mobile service unit, which were provided by the Robotics Research Lab[2]. The result is deployed in an assisted living environment, which was established during several projects at the Fraunhofer IESE [3] [4].

The outline of this paper is as follows: In Section II, some architectural approaches in other projects relating to intelligent environments are listed. In Section III, AmICom and its API is described in detail, with focus on the requirements of ambient intelligence. We continue with a description of our assisted living demonstrator and the services and applications relevant to this work in Section IV. We conclude by discussing the implemented features in AmICom and our experiences with implementation and integration of services. Furthermore a discussion on additional features which may enrich the powerfulness of our middleware within the current system architecture is given.

[1] **http://vs.informatik.uni-kl.de**
[2] **http://rrlab.informatik.uni-kl.de**

II. Architectures of Ambient Environments

Multiple research groups address the field of intelligent indoor environments. The Amigo-project at Fraunhofer IPSI in Darmstadt offers functional architectures for several high-level parts of an ambient system called intelligent user services. Context management, human profiling, notification and user interface services are addressed. One distinguishes human-oriented and system-oriented services thus adapting the service behavior towards the human user [5]. The intelligent room project at MIT uses a three-layer architecture for sensor data interpretation, abstraction and application [6]. Objective is supporting humans by interpretation of their actions, gestures, and speech. The AwareKitchen at TU Munich uses the robotics-inspired Player-Stage platform [7]. In [8] we described the architectural and communication-oriented capabilities of the robotics framework that so far was in use in our applications, the Modular Controller Architecture (MCA) [9] and how it could be used in ambient intelligence systems. Especially the shared memory approach of MCA offered a great possibility to facilitate distributed dynamic applications. However as also described in [8] MCA lacks of certain features regarding ad-hoc communication that are mandatory in ubiquitous environments. The Aware Home in Atlanta uses generic service components called widgets that communicate over the HTTP and the SMTP protocol with XML style messages. Generally, one tries to avoid CORBA or RMI due to their heavy weight [10]. The I-Living architecture is being developed for assisted living appliances at the University of Illinois. The system is centralised and uses a set of gateways for accessing proprietary hardware. A focus of the system lays on robustness through redundancy. For example if the land-line internet access fails, the system automatically activates and switches to a cell-phone connection [11].

In general, three types of communication architectures can be identified: message-based, RPC, and shared memory. Besides the development of real-world demonstrators where usually straight-forward approaches are followed, a number of researchers address primarily the communication or service-interaction aspects of pervasive systems without specific mappings to appliances [12]. Again, the interesting statement is that remote procedure calls should be avoided in ambient environments as they rely on static interfaces and stable networks [2]. A widely used framework for home environments is the OSGI framework. Ignoring the fact that OSGI is only available for Java, there is still one conceptual problem: The centralised architecture with a home server (residential gateway) stands in contradiction with the ubiquitous paradigm of a distributed service federation.

III. The AmICom Middleware

In this paper, we introduce *AmICom*, the tailored lightweight communication middleware for ambient intelligence networks, which is based on distributed services. Each application in the network may register services, which are then available to all nodes in the network. Thereby, all other applications can subscribe to this service to form a multicast communication group.

Ambient intelligence environments consist of different type of nodes, which communicate through wired networks or by wireless mobile ad-hoc networks that come into existence by the mere presence of nodes that form a self-organizing network. They support distributed user applications in various areas of ambient intelligence, including professional work, leisure activities, public health, and transportation. For these user applications, it is essential to provide their distributed services independently from the current network architecture and without reconfiguration of the network.

A. Architecture

Figure 1 shows the general architecture of each node in the ambient environment. Depending on the available hardware resources, there are possibly several applications, which may register or subscribe to a particular service in the network. For each request with an new service name, a *ServiceProvider* or a *ServiceUser* is created, representing a communication endpoint for the application. The communication between a ServiceProvider and all of its subscribed ServiceUsers is observed in order to detect the loss of communication or the failure of a node in the network. In case that the failure cannot be recovered, the subscribed applications are notified.

Additionally, the AmICom consists of routing and MAC protocols that were developed in our group[13]. They address the special challenge in mobile ad-hoc networks and facilitate the abstraction from the heterogeneous topology. Due to the low level access to several communication devices on different platforms, the communication is realised independently from the used hardware (e.g. no IP address has to be configured) and without intervention from the application developer.

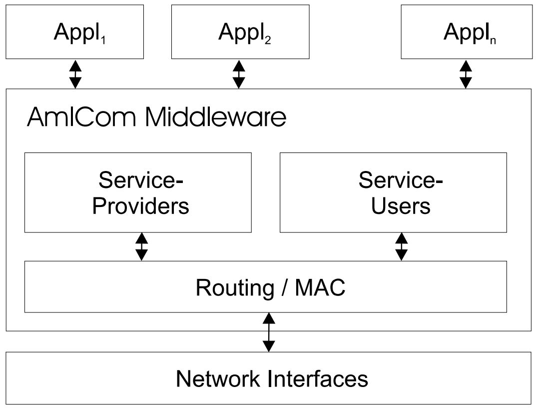

Fig. 1. AmICom architecture

B. Services

One objective of AmICom is to keep the programming interface as simple as possible to facilitate the familiarization with the interface, increase the acceptance and allow the portage to multiple hardware platforms. The API to the AmICom middleware is realised by C functions or as methods of a class in all object-oriented languages:

- bool: Register(string: name)
- bool: UnRegister(string: name)
- bool: Subscribe(string: name)
- bool: UnSubscribe(string: name)
- bool: Send(string: name, bytes: data)
- bool: Receive(string: name, bytes: data)

Figure 2 shows a typical scenario for the use of the AmICom API. The Register call (respectively UnRegister) allows an application to register a service in the AmI network with a given name. Each application interested in this particular service may subscribe to this service with a Subscribe(name). AmICom observes these operations and gives a feedback, whether the request could be fully performed. The Register may fail if a service with this name is already been registered in the network. The Subscribe fails, when the requested service is currently not available.

All applications that have registered or subscribed may Send messages that are transmitted to all other associated application with this service. These messages can be consumed by a Receive call. Additionally, the registration of callback functions is possible, allowing an immediate reception of messages and the asynchronous notification of service failure.

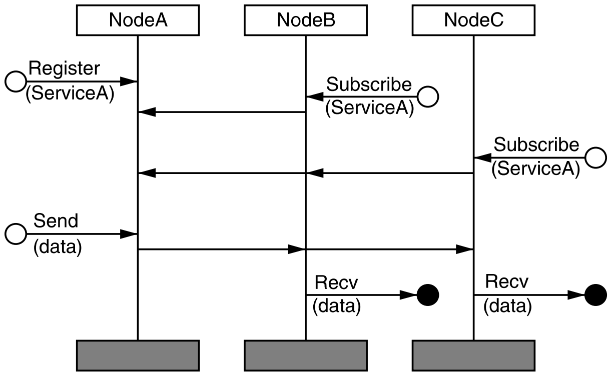

Fig. 2. AmICom: communication scenario

C. Messages

Data transmitted by AmICom on the communication level is an array of raw byte. However, on a higher level we already have a mechanism integrated to serialise data structures platform-independently. Therefore, we follow the ASN.1 (Advanced Syntax Notation) standard [14] using an interface description language with suitable ASN.1 compilers for Java, C and C++. This mechanism is an efficient and intuitive way to serialise data for transmission between different software and hardware platforms. The interface description looks similar to the one used by CORBA, while the usability is much simpler and flexible, due to the modularisation and the availability of free libraries for different platforms. Additionally, the developer may choose different kinds of encoding and decoding algorithms that all consider the byte order of the transfered data among the different hardware platform.

The description of data structures as depicted in an example in Section IV-A are compiled to C structs and Java classes. During runtime, the applications may allocate corresponding objects, fill them with data and encode them before transmission. On the receiver side, an object of the expected type is created and filled with the raw data. The corresponding decode method returns successfully if the object accepts the data.

D. Platforms Nodes and Tools

One important design criterion for AmICom was interoperability. Therefore, the core library facilitating the API described in III-B is accessible in C, C++ and Java through native interfaces. The complexity of this administrative communication interface is low enough to be ported even to small hardware nodes like Particle Computer or MicaZ nodes. Here, a gateway for the physical communication layer is necessary. These gateways have to translate all AmICom messages between the physical networks. However, it is important that service interactions from a developers point of view are transparent between nodes connected to different networks. Communication between nodes in wireless and wired networks is realised by either a WLAN access point or through multiple network interfaces per node. So far, we did not develop gateways for smaller nodes but this is considered as one of the next major steps in AmICom development.

IV. Ambient Assisted Living Demonstrator

The demographic development is of great concern in industrialized countries as discussed in recent works. The fear about the negative social and economical impact triggers many research initiatives that investigate different kinds of approaches. One idea is to reduce the impact by enabling elderly people to remain in their familiar living environment instead of moving to nursing. Development of technology in terms of ambient hardware and intelligent software is seen as one possible approach. The participating research groups within the BelAmI Project [15] focus on energy-aware electronics, QoS-communication, software engineering and adaptivity, safety and security, human-machine interaction and mobile robotics. Numerous research results and technological achievements have been already installed in our research lab.

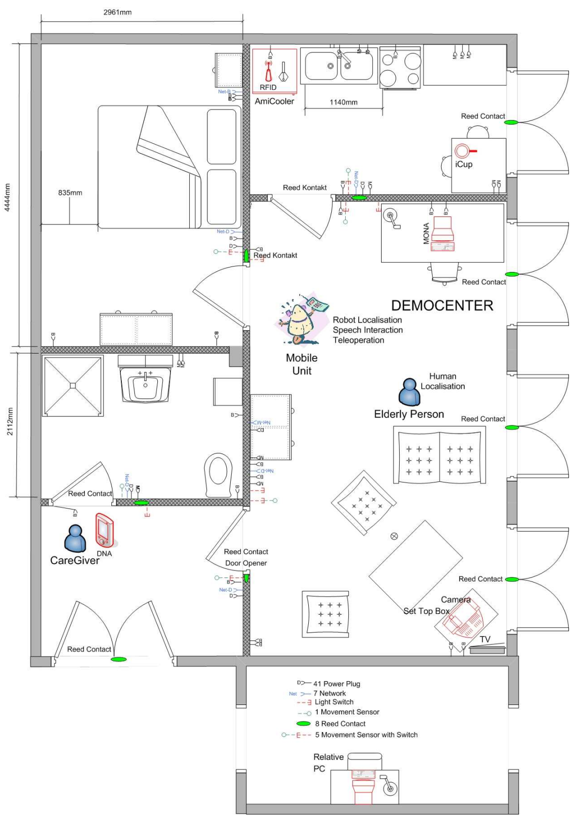

Fig. 3. Demonstrator Setup

The demonstrator, shown in Figure 3, contains a full $60m^2$ living environment with bedroom, bathroom, living room and kitchen. This Section describes the demonstrator setting regarding several applications which have been implemented on top of the AmICom middleware so far. These applications include human-machine interaction and the control of the mobile unit as described in the following Sections. The applications have to deal with data interpretation on the semantic level, but this can never been avoided by platform mechanisms.

A. Display Service

We have attached several displays for text output to our nodes. The interface of the according service includes a display command of specified text messages, which are displayed for a certain amount of time. Below, the interface description in ASN.1 is presented, which is given to the compiler as described in Section III.

```
IfSerialDisplay
DEFINITIONS ::=

BEGIN
IfSerialDisplay ::= SEQUENCE
  {
     clear BOOLEAN,
     time INTEGER,
     text PrintableString
  }
END
```

B. Audio Player Service

For this service to work, nodes have to feature multimedia audio capabilities. Another application may subscribe to this service and invoke the playing of a specified audio file.

C. Speech Dialog Service

In [16] a speech dialog engine to support human-system interaction has been developed. A mobile robot or a stand-alone PC with microphone and speakers is able to play the role of a transactive interaction agent for the whole ambient system containing various heterogeneous nodes of different computational capability. The advantage is two-sided. On the one hand, the user only has to address one artificial communication partner. On the other hand, also services running on low-power nodes are enabled to participate in the interaction by speech. Their interaction description is specified in XML, provided through an AmICom message. The message is processed by the speech engine on our robot, described in the next section. We believe that using a small mobile robot as interaction partner emerges a higher user-acceptance than distributing numerous microphones and cameras within the living environment.

D. Services of the Mobile Unit

The mobile unit ARTOS (Autonomous Robot for Transport and Service) has a length of 50 cm, a width of 30 cm and a height of 25 cm. It is suitable for indoor application especially in home environments. ARTOS is a hardware node in terms of this work. It features the already described multimedia services with additional facilities for interaction, tele-operation for emergency recognition, transport services and human-behaviour monitoring for health supervision. The robot carries a speech engine, mentioned in Section IV-C to perform transactive interactions which enables the robot to become an interaction agent for various devices and services in the intelligent environment and be the mediator towards the human. The internal control software of ARTOS regarding drive control, anti-collision sensors, human-detection sensors, interaction multimedia, and camera-vision are developed in MCA. The interfaces, which are accessible over AmICom, include among others the robot's location and sensor states. The robot also offers interfaces for delivery of speech and text messages. Artificial landmarks using a grid of passive RFIDs in the floor (see Figure 6) realise the localisation. The robot uses the inhabitant's location information which is provided by the sensors.

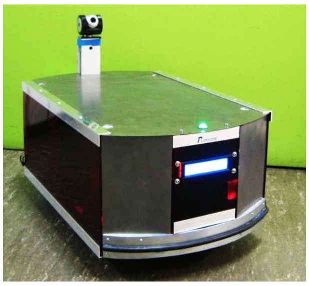

Fig. 4. ARTOS

E. Fridge Monitor Service

The assisted living demonstrator addresses health monitoring and among others the supervision of nutrition and liquid consumption. We contributed with an intelligent refrigerator using RFID-tagged food items to report expiry dates. Just like the ARTOS node, the refrigerator features an audio player and a display service. Warning messages that the food has expired are sent to all available display services in the network.

F. Human Location Service

For certain services, it is useful to know the location of the inhabitants. Especially for elderly people in assisted living environments these location-based services can be used to enhance the service of applications. For example, the position of the human can be published to the mobile unit, described in Section IV-D, so it can offer services and interact more precisely. We developed two prototypes of localisation systems based on active and passive RFID technology, which are integrated via AmICom. A foot-mounted RFID sensor in Figure 5 provides localisation information with a precision of about 15 cm. if the shoe is near to the floor. In other cases, an additional sensor communicating with active RFIDs in the ceiling steps in. This second solution offers only room precision but can easily be scaled to larger buildings without expensive infrastructure.

Fig. 5. Foot-mounted Sensor

Fig. 6. RFID Underlay

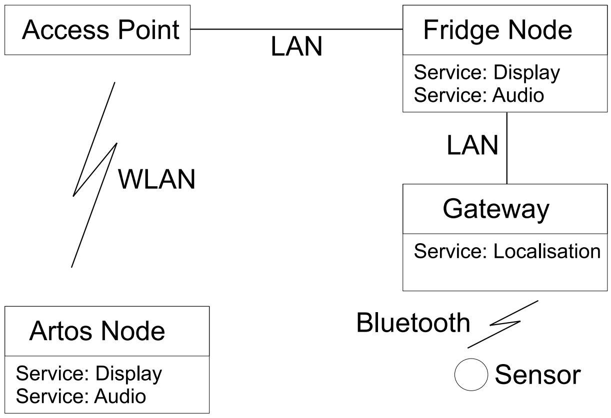

Fig. 8. AmICom: Collaboration Example

G. Video Telephony

The telephone is a communication media that helps to maintain friendships, but is reduced to voice, while a video telephone enables people to see each other and offers new possibilities. People can talk to friends, family members, but can also use it to talk to their doctor. In case of an emergency the video telephony registers services for audio and video communication in the AmI network allowing assisted persons to communicate with a predefined person.

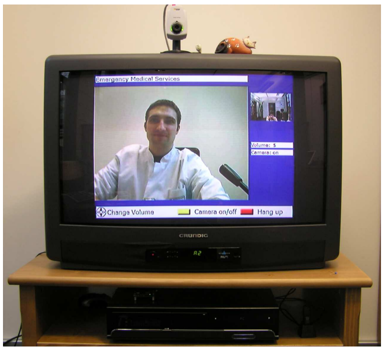

Fig. 7. Video Telephony

H. Active Badge

The Active Badge is a small device the assisted person carries around. The badge has an emergency button attached, which activate an alarm in the AmI network. Additionally, the pulse of the person is observed and may possibly trigger the alarm autonomously. The registered alarm service in the AmICom can be used to establish a video telephony connection to the doctor, who can retrieve additional information, as e.g. the heartrate.

I. Scenario

Figure 8 shows an exemplary scenario that consists of several of the previously listed nodes and services. Here, the inhabitant is warned about expired food in the refrigerator.

The intelligent refrigerator - as source of information - subscribes a display service and an audio output service running on the same node. In addition, the human location interface and the display on the mobile Robot ARTOS are subscribed. If a food warning has to be delivered, the refrigerator service checks if the human inhabitant is located within a certain vicinity. In the case the human is present in the kitchen, the refrigerator uses the locally available output devices for the warning messages, while in the other case, the mobile unit is used. A software developer has to anticipate the possible settings here. In a future release of AmICom wildcards in service names become available, which would allow sending messages all displays available in the network, regardless on which node they are deployed.

V. Conclusion and Outlook

The previous example shows the possibilities that emerge from a communication system that only depend on distributed service with names. Furthermore, this communication is realised without any centralised node or service and without the necessity to deal with any network details. For truly ubiquitous service interaction, the AmICom middleware offers a fully distributed concept for message transfer. Services are registered and subscribed by use of qualified names. All network related configuration parameters are hidden from the application developer. By using the ASN.1 scheme, a powerful and unified mechanism is integrated to serialise, encode and decode arbitrary data structures with certain level of type safety and data accuracy. The asynchronous messaging principle allows intuitive collaboration of ambient services in our assisted living environment.

A. Integration

We described several concrete applications that were implemented using the AmICom middleware and integrated within our assisted living demonstrator. Especially the independence and automatic service (re)connection happened to be extremely helpful during development and test of application functionality. Shutting one component down, activating another, replacement of nodes etc. has been a great deal of work so far, since previous solutions often required a full restart of all nodes.

B. Future Work

Besides the advantages of AmICom regarding the communication layers, it is now time to begin with adding features for convenience. Watchdog functionality for services and nodes, wildcard service loop-up and transparent integration of different (smaller) hardware nodes will be the main topic for the next months. In addition, no access control is handled by the middleware yet. For example, one could think of services that can only be subscribed by one other service at a time.

VI. Acknowledgments

We gratefully acknowledge the funding of the BelAmI project by the State Ministry of Science, Education, Research and Culture of Rhineland-Palatinate, by the Fraunhofer-Society and by the German Federal Ministry of Education and Research.

References

[1] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, pp. pp. 94–104, 1991.

[2] U. Saif and D. Greaves, "Communication primitives for ubiquitous systems or rpc considered harmful," in *21st International Conference of Distributed Computing Systems ICDCS (Workshop on Smart Appliances and Wearable Computing)*, Phoenix, Arizona, USA, April 16-19 2001.

[3] J. Nehmer, A. Karshmer, M. Becker, and R. Lamm, "Living assistance systems - an ambient intelligence approach," in *Proceedings of the 28th International Conference on Software Engineering (ICSE)*, Shanghai, China, May 20-28 2006.

[4] M. Becker, E. Ras, and J. Koch, "Engineering tele-health in the ambient assisted living lab solutions in the ambient assisted living lab," in *21st International Conference on Advanced Information Networking and Applications (AINA)*, Niagara Falls, Canada, 2007.

[5] C. Magerkurth, R. Etter, M. Janse, J. Kela, O. Kocsis, and F. Ramparany., "An intelligent user service architecture or networked home environments." in *2nd International Conference on Intelligent Environments*, Athen, Greece, July 5-6 2006, pp. 361–370.

[6] R. Brooks, "The intelligent room project," in *Second International Cognitive Technology Conference (CT)*, Aizu, Japan, 1997.

[7] M. Kranz, R. B. Rusu, and A. Maldonado, "A player/stage system for context-aware intelligent environments," in *Workshop on System Support for Ubiquitous Computing (UbiSys) at the 8th Annual Conference on Ubiquitous Computing (Ubicomp)*, Orange County, California, USA, September 17-21 2006.

[8] J. Koch, M. Anastasopoulos, and K. Berns, "Using the modular controller architecture (mca) in ambient assisted living," in *3rd IET International Conference on Intelligent Environments (IE07)*, Ulm, Germany, September 24-25 2007.

[9] K. U. Scholl, J. Albiez, and G. Gassmann, "Mca- an expandable modular controller architecture," in *3rd Real-Time Linux Workshop*, Milano, Italy, 2001.

[10] A. Dey, G. Abowd, and D. Salber, "A context-based infrastructure for smart environments," in *1st International Workshop on Managing Interactions in Smart Environments (MANSE)*, Dublin, Ireland, 1999.

[11] Q. Wang, W. Shin, X. Liu, Z. Zeng, C. Oh, B. K. Alshebli, M. Caccamo, C. A. Gunter, E. L. Gunter, J. Hou, K. Karahalios, and L. Sha, "I-living: An open system architecture for assisted living," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Taipei, Taiwan, October 2006.

[12] C. Mascolo, L. Capra, S. Zachariadis, and W. Emmerich, "Xmiddle: A data-sharing middleware for mobile computing," *In Personal and Wireless Communications*, vol. 21, no. 2, April 2002.

[13] I. Fliege, A. Geraldy, and R. Gotzhein, "Micro protocol based design of routing protocols for ad-hoc networks," in *Proceedings of Notere 2007*, Marrakesh, Morocco, June 4-8 2007.

[14] "ASN.1 consortium," http://www.asn1.org.

[15] "Bilateral german-hungarian collaboration project on ambient intelligent systems," http://www.belami-project.org.

[16] J. Koch, J. Wettach, H. Jung, and K. Berns, "Dynamic speech interaction for robotic agents," in *13th International Conference on Advanced Robotics (ICAR07)*, Jeju, Korea, August 21-24 2007.