AwareCon: Situation Aware Context Communication

Michael Beigl, Albert Krohn, Tobias Zimmer, Christian Decker, and Philip Robinson

TecO, University of Karlsruhe, Vincenz-Priessnitz-Str. 1
76131 Karlsruhe, Germany
{michael,krohn,zimmer,cdecker,philip}@teco.edu

Abstract. Ubicomp environments impose tough constraints on networks, including immediate communication, low energy consumption, minimal maintenance and administration. With the AwareCon network, we address these challenges by prescribing an integrated architecture that differs from classical networking, as it features an awareness of the surrounding situation and context. In various settings, where AwareCon was implemented on tiny battery driven devices, we show that applications and usability of devices benefit from this approach.

1 Introduction

Communication between (small) embedded computing and sensing devices is an integral facility in Ubiquitous Computing scenarios. It is often the case that existing communication technologies, such as wireless LANs, mobile phone networks, or standards for wireless personal area networks such as Bluetooth or IrDA, form the communications backbone of these scenarios. However, it is our experience that these networks lack a significant measure of situation adaptability and are optimized for particular settings, not necessarily representative of typical Ubicomp application scenarios. For example, WLAN has increased in popularity as a standard for data transfer in offices and homes, while Bluetooth and IrDA are convenient replacements for cables between devices in close range. However, Ubiquitous Computing settings are characterized by a greater variance in networking ranging from sparse communication out in the field, where devices may come together infrequently, to dense communication environments consisting of possibly hundreds of devices in one room.

In this paper we propose a communication system that is designed to adapt to various settings by generating and using context and situational information for improving the overall system performance. In Ubicomp, context or sensor data is often used as input for applications, which run as a layer on top of a network stack like in TinyOS [1] and ContextCube [2]. Our design and implementation of an Aware-of-Context (AwareCon) – network, shows how a network could benefit from intrinsic processing of context information rather than simply acting as its transport medium.

Existing networks for small, embedded devices often emphasize particular (context) implementation issues. For example, the Prototype Embedded Network (PEN) [4] provides advanced energy saving by using very simple and passive devices. Blue-

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 132–139, 2003.

tooth [5] and other standard bodies in the domain of ad-hoc networking (e.g. IEEE 802.15 [6] (esp. TG2) and ZigBee [7]) currently have no built-in notion of context awareness in their protocol stacks.

Today, the development of context aware systems such as Toolkit [8] and TEA [9], build on top of the communication subsystems rather than allowing a measure of integration. Our contribution with AwareCon functions as a bridge between situation aware systems and network design in Ubicomp.

In Ubicomp settings, an assortment of devices works together to unobtrusively support the human through response to explicit human computer interaction, and, more significantly, rapidly changing context. We concur with other researchers (e.g. [10]) that unobtrusiveness is a core operational feature of many technical systems that are intended for background environmental augmentation. Further challenges include low power consumption (e.g. [11]), immediate communication between unknown devices, robustness and no system administration (e.g. challenges 2,3,6 for Ubicomp in homes from [12]). From our experience with systems and settings requiring non-stop operation (e.g. MediaCup[13], AwareOffice [14] with several years of operation), we found frequent maintenance to be particularly inconvenient, and suggest that minimization of the maintenance effort is an essential and practical requirement for Ubicomp devices.

2 Context and Situation Aware Networking

Meeting the above challenges with strictly classical networking approaches has proven to be an arduous undertaking. We therefore claim that *situation awareness is essential for Ubicomp devices and their communication, to meet acceptable performance efficiency*. This concept underpins the AwareCon communication system. Each instance of the network (realized as small, mobile devices) must be able to produce, store and consume relevant information.

In situation aware networking we use context to represent situations. The situation of an artefact (a mobile device) is a collection of context information that leads to adaptive decisions, including communications behavior. Context in Ubicomp settings is derived from environmental information, obtained from sensor data, and from meta-information of the application domain, generated by a *context producer*. Context models like the one proposed by Dey, Abowd and Salber in [15], which focus on how to handle the "content" of the context, do not cope with its underlying structure. In contrast, we concentrated on the structural attributes of context and identified validity, relevance, reliability and context history as the most influential properties. They define the representation of context in AwareCon. Although AwareCon is implemented as a complete network protocol stack, it has unique features for context aware communication. Context information is contained within the payload of communications packets. Nevertheless, it is still possible to transport any other raw data on higher protocol layers.

Context is thus used in two ways in the proposed system: Firstly, context data for use by applications is encoded within the payload of communications packets. Secondly, situational context information used for network optimization is processed at separate component of the protocol stack.

3 The AwareCon Network Stack Architecture

The AwareCon protocol stack provides typical network features for coding, access and transport. The AwareCon protocol is divided into layers and components. This structural approach is a well-founded principle in network engineering.

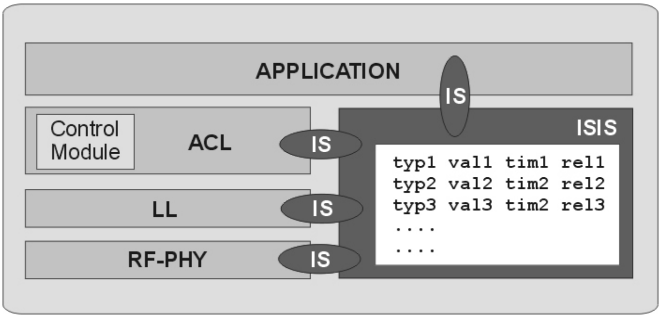

Fig. 1. AwareCon Protocol Architecture

The basic architecture consists of 5 components (Figure 1). Four of the components are traditional communication layers: the Radio Frequency Physical Layer (RF-PHY), Link Layer (LL), and Application Convergence Layer (ACL), with a control module for remote administration and the Application Layer. In addition to the traditional network layers, we have introduced a component called the Internal Situation Store (**ISIS**), with accompanying Interpretation Stubs (**IS**) at each of the traditional layers. The ISIS is the core, facilitating component for context awareness as it holds all information relevant to the context-based enhancement of the communication and application. Contexts can originate from internal or external sources. Internal sources include functions that interpret status variables of the protocol stack or sensor values, whereas external sources are e.g. remote devices or service points that broadcast context information. Data elements stored in the ISIS are clearly separated from the actual payload data that is transported for application purposes.

In the style of the earlier identified attributes of context, we decided to implement the following 4 attributes for contexts stored in the ISIS: *type* of situation, *value* of situation, *time* stamp of last change and *reliability* of the value. The Interpretation Stubs are used to push situation data (context) of interest into the ISIS, and subsequently to interpret outbound data for the respective part of the system and provide an easy access interface. Theoretically, any context and situational information could be stored in the ISIS. At the current stage, the generation and storage of the following contexts are implemented:

- energy resources (battery level)
- processor load (percentage of busy time)
- link quality (bit error rate/packet error rate)
- number of active devices

These values are produced and consumed for improvement of the performance of the network stack and applications. In the next section, we take a closer look at three of the mentioned context values in the ISIS. We explain where they are generated and how the consumption of that information improves the network or application.

4 Applications Based on AwareCon

For some time now we have constructed various Ubiquitous Computing environments and applications, which have been intermittently running for several years. The AwareCon network is an outcome of collective findings and experience made with these settings. At the same time, we also use the settings as test-beds for evaluating the functional features of the AwareCon. AwareCon is implemented on TecO's generic hardware platform (see Figure 2) designed and built during the Smart-Its project [16]. Smart-Its are very small computing, sensing and communication devices that are especially designed for post hoc augmentation of everyday objects. In the following sections we also refer to Smart-Its as artefacts.

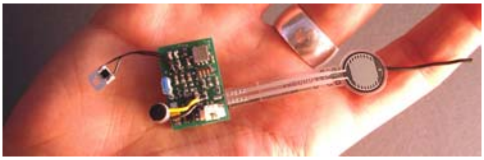

Fig. 2. TecO's Smart-Its: A general sensing, computing and communication platform

4.1 Energy Saving for AwareOffice

Long-term operation of computerized artefacts is important on the Ubicomp agenda, as was mentioned at the beginning of this paper as one of the main network challenges. AwareCon exploits context awareness as a means of *energy saving*.

The "chair appliance" in the AwareOffice scenario is a meeting room chair with a Smart-It device attached. These were our primary implementation and evaluation test beds for the power saving mechanisms. In this setting, two contexts stored in the ISIS inform the energy saving process: *the number of active devices* - a context produced in the network stack (RF Layer) and the *remaining energy* – a local sensor value available on each mobile device. The chair appliance interprets certain movements of a chair to resolve that someone is sitting or that the chair is being moved. The detected status information is then communicated to other artefacts in the environment via the AwareCon network. The chair shows typical communication behavior for electronic artefacts in Ubicomp: the communication is sparse, and unpredictable. The transmission of outbound packets results only from user interaction.

Chairs – here as example of everyday objects equipped with electronics – are typically unsupervised and are seldom or never maintained, such that some consideration is necessary to achieve long-term operation of the electronics (reduce mean power consumption). Consequently, the major contributor to energy consumption was the communication. Without human interaction, the chair's Smart-It is in sleep mode, consuming minimal power. Upon movement of the chair, the attached Smart-It starts its processor and communication, sends out a packet containing the actual movement state or pattern and returns to sleep mode. Running the processor and protocol stack consumes around 100 times more power than the sleep mode. Therefore it is a clear goal to minimize the "up time" of the chair's Smart-It. The mean time for sending an outbound packet has a significant impact on the over all mean power consumption. The longer the send operation has to wait for channel access, the more energy has to be spent waiting with the processor and protocol stack running. An arbitration in the Link Layer resolves the distributed transmit inquiries. The mean delay time until a packet is transmitted depends on the number of active, continuously transmitting devices. In this circumstance, context awareness suggests a prioritization of only those artefacts (in this case the chair appliance) that are low on energy resources, to reduce the delay for sending of outbound packets. Assuming various types of artefacts in one scenario, only those with critical energy resources (e.g. depending on batteries) will invoke the mentioned energy saving mechanism. To control the collision rate, the knowledge of the number of active devices is necessary to adjust the arbitration for the channel access.

With a mean of 1% power-up quota (99% sleep mode) for the chair's Smart-It, around 100 days of operation can be achieved with a 500mAh battery. Giving the outbound packets of the chair a higher priority could easily shorten the mean delay for packet delivery by a factor of ten. This results in a longer lifetime of around 200 days (ignoring the self discharge of the battery).

A predictable lifetime of an artefact until replacement or recharge of batteries is another issue. As stated, the power consumption of the chair's Smart-It depends strongly on the activity that takes place with the chair. This will result in different battery cycles due to different activities. However, for administration and maintenance it's important and helpful if no artefact ever runs out of energy before the predicted lifetime.

Therefore, the network protocol stack can react according to the *energy resources* context of the ISIS. In a situation where Smart-Its are low on remaining energy resources, they can reduce their transmit power to save energy and invoke a simple repeat request (flooding) of their packets to transport them further. While transmitting, the transmit power is more than 50% of the whole power consumption of the artefact. Transmitting with minimum power and a 1% power-up quota can reduce the mean power consumption by 30%, implying 30% longer lifetime after starting the mechanism. In doing so, Smart-Its use energy of the surrounding devices – functioning as transmission repeaters - to equilibrate the energy consumption of artefacts in one area.

4.2 Number of Active Devices: Application Triggering

Some applications depend on partner devices in a certain area. The absence of these partner devices would shut down the applications. RELATE, a 2D surface location system for tangible objects [17], is one example built by us using Smart-Its as communication technology. RELATE enables the distributed localization of objects e.g. on a white board with no infrastructure, and is therefore an excellent Ubicomp example scenario.

The ISIS value *number of active devices* (produced in the RF Layer) is used to determine if possible partners are around. The RELATE technology uses infrared signals to determine location. The scanning and localization of objects requires certain effort in computation and energy. Therefore, it is essential to avoid invoking the location mechanisms when no partner devices are around.

With the knowledge of the *number of active devices,* it is further possible to influence the update rates of the localization algorithm. This is of special interest for RELATE because the human interaction with objects carrying RELATE functionality should be supported and enabled in real time, which makes scaling and update rates critical issues.

4.3 Computation Time Prediction for Context-as-a-Key

Small embedded devices like Smart-Its and Smart Dust [3] are often single processor solutions. This reduces the complexity and the energy consumption, and simplifies the development process in the research stage. Computation time spent for the networking on such a single processor design can consume a substantial part of the overall available computation time. This may lead to conflicts with certain application tasks - e.g. digital signal processing - that are also in need of great amounts of processing power. In real time applications computation has to be finished after a certain time frame. With high network activity, this real time behavior might not be reached.

The AwareCon stack provides the context *processor load*. This information – produced in the RF layer – can be used by the application to control its own run time behavior. *Context-as-a-Key* [18], a security service for mobile devices, provides an encrypted communication based on symmetric keys generated from a common context. Devices in close range use synchronously sampled audio data from their environment to generate these keys. The key generation and encryption – running on one processor with the network stack - demands high computation effort. Depending on the known *processor load,* the security service is able to predict its response time in advance and the application can calculate the possible frequency of secure communication. This enables adaptive behavior of time critical applications.

5 Needed Technical Characteristics of the AwareCon Stack

As shown in the above applications, it is necessary to produce certain context values to implement the context awareness of the network system. These values must be generated reliably and fast. The context awareness seriously depends on the performance of the production of the necessary contexts. AwareCon provides certain features to generate several contexts quickly and dependably. For effective situation aware communication, we summarize and amend necessary properties and features that are implemented in the AwareCon system (especially the protocol stack) on Smart-Its:

- Decentralized and cellular architecture and media access
- Ad-hoc behavior and spontaneous book-in into a network
- Real time communication and synchronization
- Predictable processor load due to the network stack
- Low power consumption
- Small package
- Easy (no) administration and maintenance

The network protocol stack addresses the typical requirements for an Ubicomp setting by providing real ad-hoc link establishment (typical 12ms), low power consumption (<10mA), various energy saving mechanisms, permanent synchronization (4us between any pair of nodes), distributed access control with predictable and adjustable collision rate, error control, over the-air configuration and programming, and addressing, yet also anonymity. Including these features, the fixed slotted AwareCon protocol achieves a data-rate of 48kBit/s in the 868Mhz band.

6 Conclusion

The features and functionalities explained position AwareCon as an example of a network architecture appropriate for typical Ubicomp scenarios. It uses situation aware communication in order to address earlier identified challenges and also enables applications to gain from the situation awareness. The examples in the previous sections showed how energy saving and application control during run time is possible with an underlying subsystem that generates context awareness. The selected values stored in the ISIS were influenced by the application and the effort to implement. These are therefore not the extent of selectable attributes and require some further assessment. Further context values in the ISIS and new ways to produce and consume them are necessary. The ISIS is meant to be a design initiative, advancing the beneficial influence of situation awareness on applications and their supporting infrastructure (hardware, networks etc.). Context aware applications can use the ISIS in order to maximize performance potential. With the AwareCon architecture, context information is available for any component of the system and opens a new playground for adaptive algorithms and applications.

References

1. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D.E., Pister, K.:System Architecture Directions for Networked Sensors. ACM Sigplan notices,v. 35(#11) pp. 93-104, Nov (2000)

2. Hähner, J., Becker, C., Bauer, M., Schiele, G.: ContextCube - Providing Context Information Ubiquitously.International Workshop for Smart Appliances and Wearable Computing (IWSAWC) 2003, Providence, RI USA, Mai (2003).
3. Pister, K.S.J., Kahn, J.M., Boser, B.E.: Smart dust: Wireless networks of millimeter-scale sensor nodes. University of California, Berkeley, USA (1998)
4. Bennett F., Clarke D., Evans J.B., Hopper A., Jones A., Leask D.: Piconet: Embedded Mobile Networking, IEEE Personal Communications, Vol. 4, No. 5 (October 1997) 8-15
5. Sonnerstam, D. (Hrsg.): Specification of the Bluetooth System Version 1.0 B, Bluetooth SIG (1999)
6. 802.15, Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs), 14 June (2002)
7. ZigBee Alliance [Online] from: http://www.zigbee.org [Accessed 04/22/03]
8. Dey, A.K., Salber, D., Abowd, G. D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. HCI, Vol. 16 (2-4) (2001)
9. Gellersen, H.W., Beigl, M., Schmidt, A.: Sensor-based Context-Awareness for Situated Computing. SEWPC00, Limerick, Ireland (June 2000)
10. LaMarca, A., Brunette, W., Koizumi, D., Lease, M., Stefan Sigurdsson, S., Sikorski, K., Fox, D., Borriello, G.: PlantCare: An Investigation in Practical Ubiquitous Systems. UbiComp 2002, (2002)
11. Smit, G.J.M, Havinga, P.J.M: Lessons learned from the design of a mobile multimedia system in the MOBY DICK project. In Proceedings of HUC2k, Bristol, UK (2000)
12. Edwards, W.K. and R.E. Grinter: At Home with Ubiquitous Computing: Seven Challenges" Proceedings of the UbiComp 2001, Atlanta, Georgia, USA, 2001
13. Beigl, M. Gellersen, H.W., Schmidt, A.: MediaCups: Experience with Design and Use of Computer-Augmented Everyday Objects. Computer Networks, Vol. 35, No. 4 (2001)
14. The AwareOffice Initiative at TecO [Online] TecO, University of Karlsruhe, Germany, Available from: http://www.teco.edu/awareoffice [Accessed 04/22/03]
15. Dey, A., Abowd, G., Salber, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. HCI, Vol. 16, (2001)
16. Beigl, M., Zimmer, T., Krohn A., Decker, C., Robinson, P.: Smart-Its - Communication and Sensing Technology for UbiComp Environments. Technical Report ISSN 1432-7864 (2003)
17. Assessment of relative positioning technologies for compositional tangible interfaces (RELATE), [available online] http://www.teco.edu/relate/
18. Robison, P., Beigl, M.: Trust Context Spaces: An Infrastructure for Pervasive Security in Context Aware Environments. Proceeding of SPC 2003, (2003)