Transport supervision of perishable goods by embedded context aware objects

A. WESSELS, R. JEDERMANN, W. LANG
Institute for Microsensors, -actors and –systems (IMSAS)
University of Bremen, Dept. 1: Physics / Electrical Engineering
Otto-Hahn-Allee, Build. NW 1
GERMANY
awessels@imsas.uni-bremen.de

Abstract: Intelligent freight objects can be introduced in order to simplify the solution of complex logistical planning tasks. This enables splitting logistical problems to local executable subtasks. The concept of Ubiquitous Computing (UbiComp) presents a model to realize such intelligent objects. A local task can be handled autonomously by a group of objects. The realization of UbiComp needs a platform that first provides the computational resources for the implementation of decision algorithms and then, secondly, the position information in order to enable context-aware features. This article presents a concept for the autonomous supervision of perishable goods and introduces the required soft- and hardware. Freight objects are represented by individual software components, which are realized with the JAVA framework OSGi. The whole software is designed to run on wireless sensor nodes to create embedded objects. Because temperature values can differ inside a reefer container, the signal strength of an RFID reader is used to provide position information by a cell based localization. A new approach is presented, where four RFID antennas are used to locate goods inside a container.

Key Words: Ubiquitous Computing, embedded Systems, JAVA, OSGi, localization, RFID, RSSI

1 Introduction

Today logistics has become increasingly demanding, it is more important than ever to guarantee the quality of goods. Especially, for critical goods like medicine and exotic fruits with long shipping times, seamless transport supervision is of high importance. According to the U.S. Food and Drug Association (FDA) 20% of all perishable food is wasted during transportation [1]. To guarantee an adequate quality it is important to monitor the whole transport chain.

During transportation high temperatures, blocked airflows, or defective seals can lead to differences in temperature values inside the container. While cooling the container, the temperature can vary in different locations of the container. These differences of up to 12 Kelvin can result in the reduction of local quality and shelf-life [2]. Several techniques of transport supervision are already described in different works of the collaborative research centre 637 of the University of Bremen and in our institute [3], [4]. These techniques will be combined into an embedded system to integrate the whole concept into the intelligent container [5].

The goal of this work is to offer an embedded platform to enable an autonomous control of the container. To reach this goal, the model of ubiquitous computing (UbiComp) [6] will be integrated into the container: each good is represented by agents, which collects all necessary measurements from the sensor network.

The assignment of freights and software will be done with UHF-RFID to guarantee seamless transport supervision. But, for a precise calculation of the quality of the goods it is necessary to know the position of these items. This work will present a new approach to locate goods in containers using the RFID technique. Another goal is to find an adequate middleware for these agents. To reduce the hardware and the energy consumption the entire software will be installed on the Imote2, which is a wireless sensor node platform.

Thus, it is possible to create a context-aware software representation of the freight [7]. In this work two key techniques are presented to realize such a system: the software representation of an agent, called the digital waybill, and the localization of the physical object to enable spatial knowledge for these waybills.

2 Concept

Typical UbiComp environments consist of three

Fig. 1: Concept of the intelligent container

fundamental components: the UbiComp devices, the network infrastructure, and the wireless sensors [8]. These three elements can be found in the intelligent container, too (Fig. 1). The sensor network collects all climatic information, which is sent to a base station. This base station is realized with the Imote2, which includes the Microprocessor Intel PXA271.
Therefore, this base station can act as an embedded System, which includes the agent platform. Additionally, it adapts automatically to the supervision requirements of the loaded items. During loading and unloading of the items, an agent or a digital waybill will be installed on the Imote2. This software representation of the freight includes a dynamic quality model to calculate the loss of quality during the transportation of the individual goods. If the quality drops below an acceptable limit, the waybill sends a warning to the systems, which in turn causes a re-planning of the route.
The assignment of goods and agents is done using UHF-RFID. This technique offers a cheap way to detect goods with a distance of up to 3m. Though the recognition of many RFID tags in a small area is no problem, the goods themselves limit the density of tags. The water in vegetables and other goods attenuates the electromagnetic wave extensively; therefore, it may lead to undetectable tags inside a palette of goods [9]. A secure detection of tags can only be guaranteed with intervisibility between the tag and the antenna.

But, in general cargo logistics, these limitations are of no disadvantage. In this kind of logistics the smallest documented piece is a package. Often, some boxes (or a secondary package) are combined on a palette in the supply chain of perishable goods and are named as one package (or tertiary package). But a package is always composed of the same [10]. So, from a logistics point of view it is sufficient to distinguish between each package (tertiary package). To assign each package to a waybill, tags have to be attached on the pallets.
Currently, some companies have already started to attach RFID tags on palettes, such as Palpool and European organization EPAL, who has started a pilot project on this topic [11],[12].

Four Deqtron antennas, placed at the entrance of the container, provide a secure detection of the tags. Despite their small size of 5x5 cm, the antennas are strong enough to transmit from up to three meters away. The Sirit Infinity UHF-RFID Reader from Meshed Systems is connected via the network to the Imote2 and signals when a transponder passes through the container entrance. The Reader is then assigned to a waybill, which is eventually sent over a local network. When the goods pass through a gate, the current status of the waybill is written to the transponder and can be read out by any UHF-RFID Reader. However, a reliable registration onto the RFID transponder cannot be guaranteed for all conditions. Therefore, the current status is additionally transferred via a local network. When the goods are loaded into another container, the latest status is transferred along with the waybill.
By doing so, the legally mandated supervision of perishable food stuff is made possible, and the waybill becomes a loyal travel companion of the goods. This leads to intelligent objects or goods, which are context-aware as described in the model of ubiquitous computing.

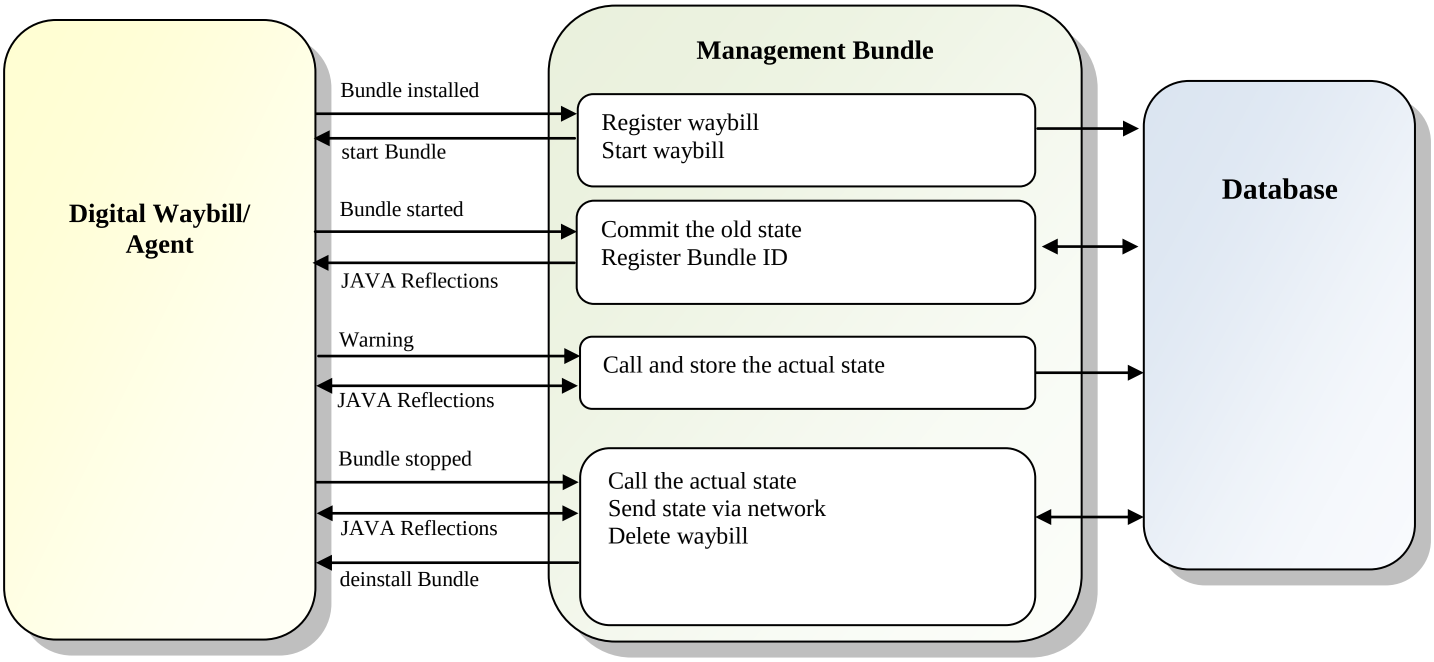

Fig. 2: Implementation of the digital waybill into OSGi

The role of the RFID gate extends beyond coordinating goods and digital waybills. It also serves to provide a rough estimation of the position of the goods. The four antennas employ RSSI values to determine if the palette is placed high or low, or whether it entered the container from the left or the right side of the entrance. In addition, the specific arrangement of the transponders allows a determination of the side of the palette that is loaded into the container. When the palette is stored on the same side as it was loaded in to the container, the exact position of the palette in the container can be determined. This position information forms the basis for retrieving the temperature records for each individual waybill.

3 Middleware

Finding the right middleware to realize this concept is one of the challenging problems. This middleware is the basis for the waybills, which defines the behavior of the whole system. These waybills will be installed when the system is running; thus, new software has to be included. As a result, JAVA is the most preferred programming language because new code fragments can be included into a running system with JAVA reflections.

First, the waybill was implemented with the JAVA based JADE [5]. This is an agent framework with its origin in artificial intelligence. However, the use of JADE involves a number of difficulties. Although JADE offers many features, the inefficient implementation of this middleware leads to long loading times for each agent. Due to the flexibility of the framework it has a very generic communication language called FIPA-ACL [13], which requires a lot of processing resources for translating objects into a formatted text message. But for an efficient implementation of the waybills, most of the FIPA-ACL's features are not necessary.
The experiments performed at our institute have shown that processing the digital waybill can last more than six seconds. In parallel and fast container loading of items into the container, this duration is unacceptable.

3.1 OSGi as standardized middleware

OSGi is the platform most suitable for this task. Originally, it was developed through a partnership between IBM, Sun, and Ericsson [14]. Today, there are many commercial and free implementations of OSGi, while Eclipse is the most known realisation. Due to its efficiency, OSGi has meanwhile developed into an industry standard of component-based programming.
The main advantage of this platform is the strict modular design of the application: with the decompression in small modules the complexity of the system gets controllable. This shortens the development of new software and relieves the maintenance of the software because the functions of each module are defined clearly. Described in 1971 by Morris the component-based engineering is nothing new [15].

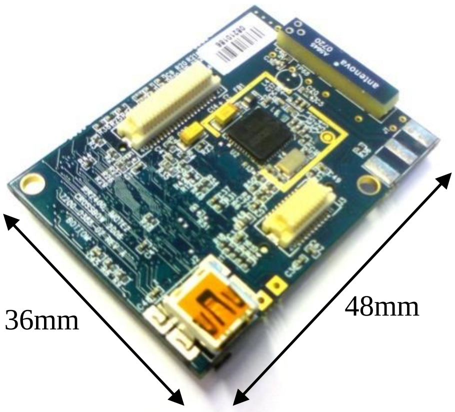

Fig. 3: An Imote2 sensor node

But, OSGi combines the advantages of the dynamic class loader of JAVA and the modular design. To control the dependencies of the OSGi modules, called Bundles, every module is loaded by its own JAVA Class-Loader. By this means the components can be strictly separated from each other, which enable the control of the dependencies of these Bundles. Furthermore each Bundle has its own lifecycle, which makes including of new modules easy [16]. Despite this module based system the inter-application communication is near-zero [17]. An OSGi Application consists completely of these Bundles.

With OSGi, self-contained "programs" can be added during runtime. Using this technique the existing system can be integrated with this new middleware. In addition, this framework offers all of JAVA's dynamic capabilities; in JADE, only one execution thread is available per agent, and the communication between separate agents adheres to a fixed pattern [18]. In contrast, OSGi supports multitasking as well as access to the classes and the interfaces of the other components.

Besides this paradigm, OSGi offers one more key advantage: it is possible to describe the communication between Bundles just by its properties [14]. This allows changing Bundles, interfaces, or hardware without changing the whole software.

3.2 Realization of the Middleware

Fig. 2 shows the implementation of the middleware using the advantages of OSGi. A management bundle stores the current state of the waybill-bundle into a database allowing seamless transport supervision. The basis for this operation is JAVA Reflections, which give access to the structure of a program. Before making an instance of a JAVA object, it is stored as Meta-Objects in the Meta-Area of the JAVA Byte-Code. JAVA Reflections are used to create a runnable instance of these objects. This powerful tool is used to load new class-files, to call methods, or even to access and set the fields of a JAVA Class. So it is possible to manipulate all variables of a JAVA Class, which makes storing of the current state of waybills possible.

This approach is used by the Management Bundle to monitor and set the state of the waybill and to store it into a persistent database. Furthermore, OSGi creates Events at each change of a Bundle's lifecycle. The Management Bundle reacts on these Events to monitor the whole life of a waybill, which makes seamless transport supervision possible.

All the communication of the digital waybills is done via events, which are described by the properties. So the whole framework can be changed during runtime when the properties of the events stay the same. Therefore, the changing of the sensors or the other periphery is easily possible during the runtime of the system. Other Bundles allow communicating via TCP/IP or acting as a web-server to get the current state of the container.

3.4 Implementation

To enable a cost efficient and space saving realization of an autonomous and intelligent container, the whole middleware is implemented on the Imote2 (Fig. 3). To run JAVA on an embedded system, it is necessary to run a virtual machine on it. Furthermore, OSGi needs a file system to install and load bundles. Thus, it is useful to run an operating system on this device.

For those reasons the embedded Linux was installed on the Imote2, which gives the capability of easy interaction with all Imote2 peripheries. Because embedded Linux is open source, it is easy to modify to specific system requirements. Therefore, it is possible to get all the data of the other wireless sensors via the RF Transmitter. In order to reduce energy consumption and costs, simple tasks inside the container were solved with the low power sensor Telosb. To communicate with the Telosb and its operating system TinyOS, the communication protocol of the Linux System was fit onto the TinyOS.

To implement OSGi on the Imote2 as efficient as possible, a small part of the code is compiled. This

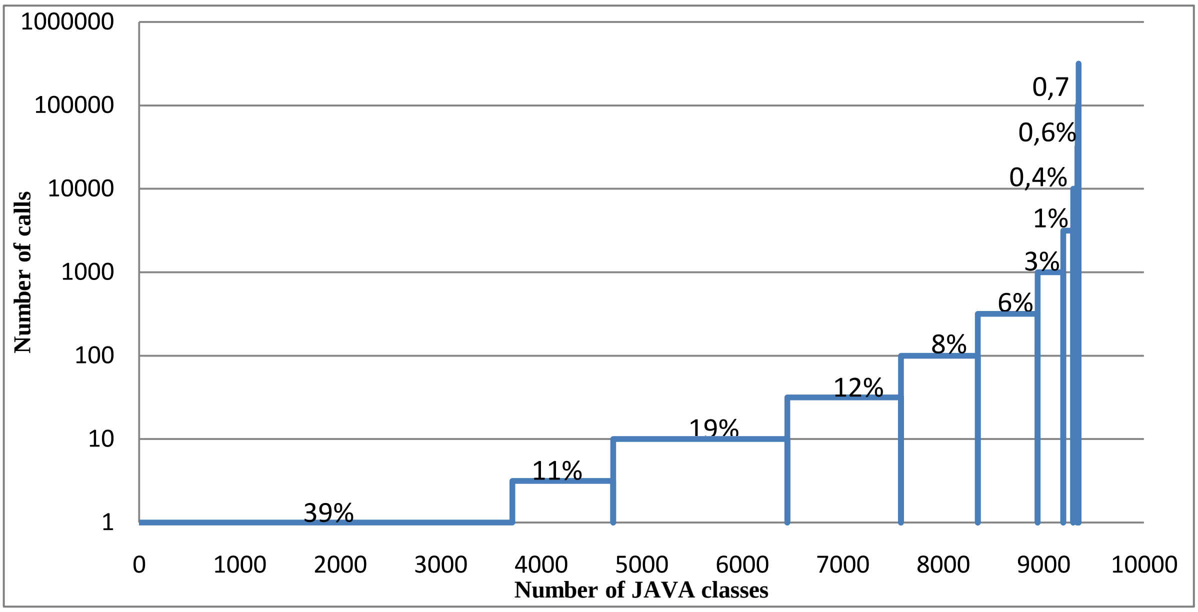

Fig. 4: JAVA method calls during a test-run of OSGi

enables a speed up in executing time compared to the interpreted JAVA code. On the other hand, the JAVA code is much smaller than the compiled code. So it was tried to use both advantages using the Jamaica compiler to get a fast and small application [19].

Fig. 4 shows each call to JAVA methods during a test run of OSGi. It is easy to see only a few methods are called very often. During this test run only 1.5% of the methods were requested 72 % of all calls.

So, compiling these most called methods would result in speeding up of the application considerably. In contrast, the program size will only increase linear. Compiling 5% of the source code reduces the start-up time more than by 30%, while the program size increases less than 6%. Thus, it is possible to create a fast JAVA application on an embedded system which uses all of the available resources.

3.5 Results

Because the entire software consists of Bundles, the modification of the entire middleware is easy. So it is possible to change sensors types or functions of the system. Only under this condition an UbiComp environment is feasible because such an environment tends to be highly dynamic and heterogeneous [20].

Furthermore, with OSGI it is possible to realize the agents or the digital waybills as bundles, which can be simply installed on and removed from a platform in less than a second, which is more than 6 times faster than JADE. The middleware allows transmitting the current state of a waybill, so that the objects become intelligent, and therefore, seamless transport supervision is possible.

In the context of supply chain, changing a platform means the shifting goods from one container into another or into warehouse. The knowledge of the position of the goods is the basis for the data processing of the waybills. But for an accurate calculation of the climatic information this knowledge is not sufficient. To consider the local differences of the temperature inside the container, the position of the goods has to be known.

4 Localization

In order to achieve an accurate localization of goods with less hardware, the position will be concluded with the spatial knowledge of the container. The basis of these conclusions is palette-wise transportations as usual in the food logistics.

This will be done like in the famous game Tetris. In this game, first objects are turned before they are falling to the final position. We assume the goods are turned in front of the container, before they are driven into the container straight ahead. Thus, the horizontal and vertical position of packages must not change during the packing inside the container.

Under this condition it is sufficient to detect the

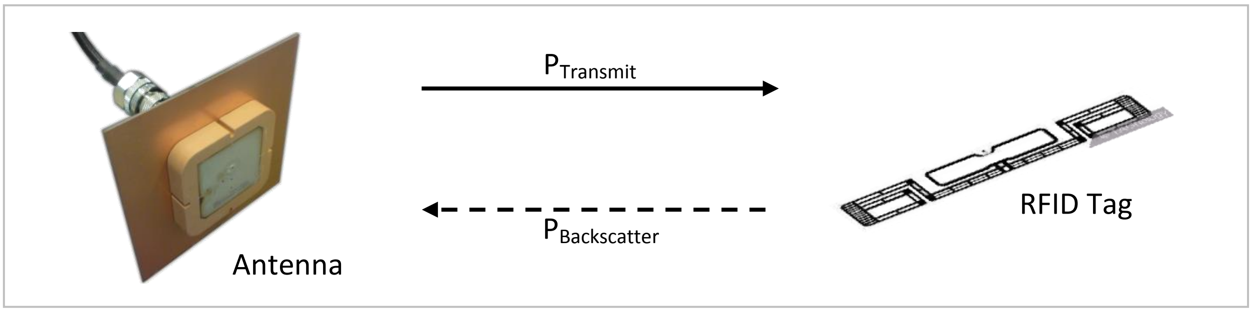

Fig. 5: Schematic of UHF RFID Backscattering

position of the goods at the RFID Gate, which is located at the entrance of the container. Knowing the history of the container and due to the fact that a container is loaded from back to the front, the final position can be extrapolated from the position in the RFID Gate.

4.1 RSSI Based Localization

The localization in the RFID Gate is based on the received signal strength indicator (RSSI) of the RFID Reader. This approach can be realized without the new hardware effort and is portable to other systems without problems.

Localization with RSSI is based on the assumption that the distance of a sender can be concluded with the deterministic damping of the electromagnetic wave in free space. However, damping of different materials and reflections can lead to large differences between the calculated and the real positions. That is why a more reliable concept for localization with RSSI is presented in the next section. To understand the special problems of this technique, it is useful to know the theoretical background of UHF-RFID.

4.2 Theoretical Background

For localization in the RFID Gate a long-range RFID system with 868 MHz (Europe) was used. One benefit of the UHF frequency of 868MHz is smaller size and better efficiency compared to low frequencies. Because of the small wave length λ of about 34cm, a UHF RFID tag is normally out of range of the electromagnetic near field.

Due to the small size of the antenna, this near field ends at a distance of

$$d = \lambda/_{2\pi} = 5{,}4\ cm. \qquad (1)$$

So, a tag cannot affect the near field of a RFID

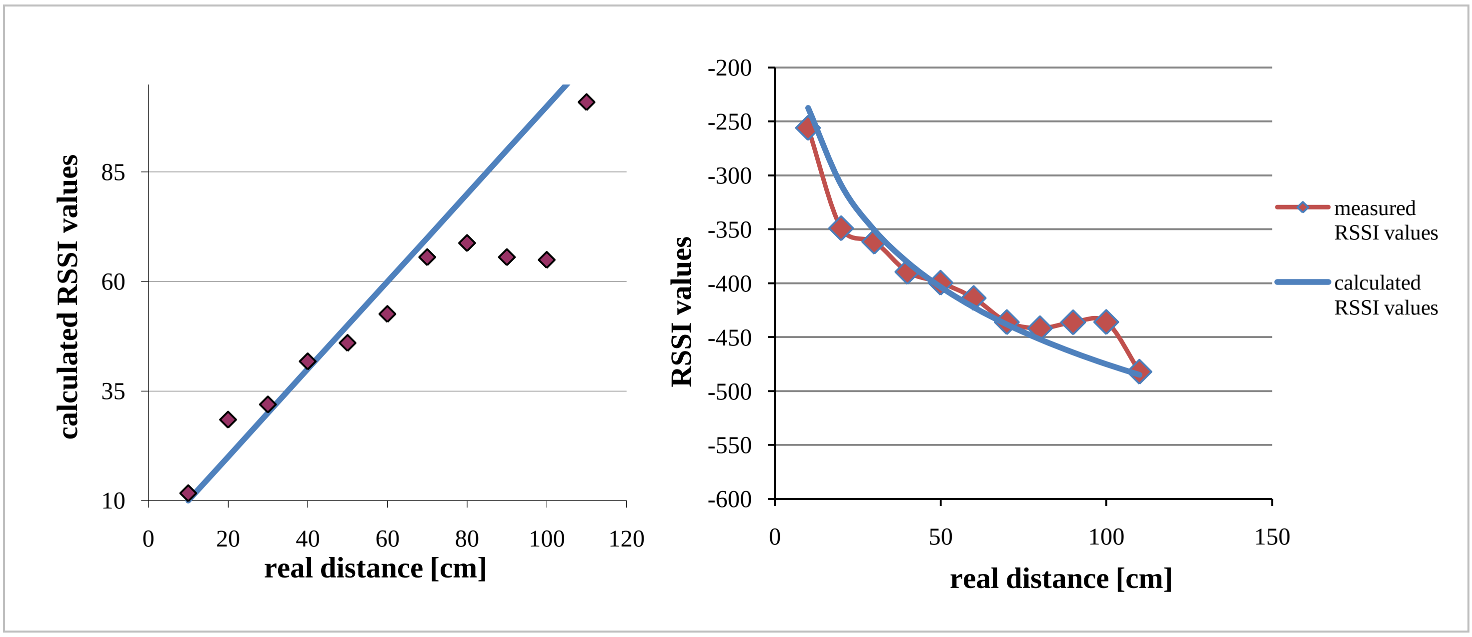

Fig. 6: Dependency of RSSI value and distance

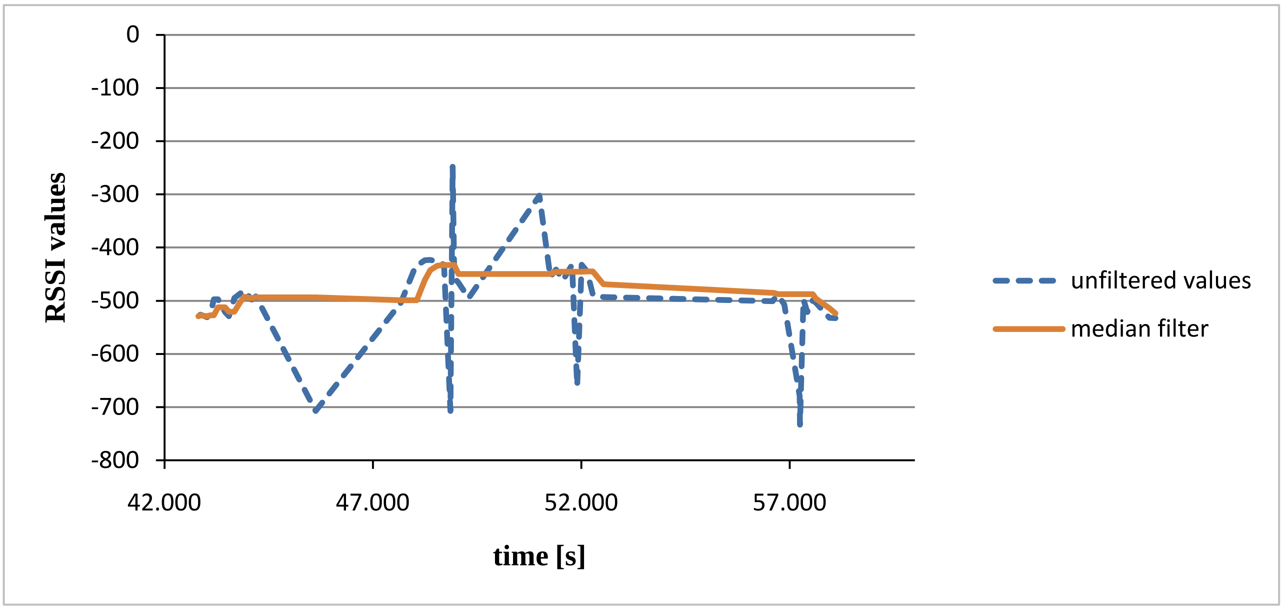

Fig. 7: RSSI values during the passing of the RFID Gate

antenna with inductive or capacitive coupling, like in LF applications. To transmit information from the tag to the antenna, the reflection of electromagnetic waves is used to transmit the data, which is called backscattering.

UHF RFID uses the fact that the reflections increase with higher frequencies. The antenna of an RFID tag is optimized to reflect the electromagnetic wave of 868 MHz (Europe). To transmit data from the tag to the reader's antenna, the antenna can change its scatter aperture σ. This is done by changing the RF impedance of the tag's antenna to control the amount of the scattered field [21]. For transmitting data to the reader's antenna, an electromagnetic wave has to propagate from the reader to the tag, which in turn reflects the wave back to the reader's antenna (Fig. 5).
For isotropic antennas the power of the reflected wave S_{BACK} can be described as follows:

$$S_{Back} = \frac{\mathrm{P_{Backscatter}}}{4\pi^2} = \sigma \frac{P_{Transmit}}{4\pi r^2} \cdot \frac{1}{4r\pi^2}$$
$$= \frac{\sigma \cdot P_{EIRP}}{(4\pi)^2} \cdot \frac{1}{r^4} \qquad (2)$$

From this it follows that the received power at the Reader will decrease with the forth root of the distance [22]. Thus, the error of the calculated distance will increase dramatically with the distance, and therefore, the localization with RSSI can be very hard.

In a wireless sensor network the performance of an indoor localization is already poor, though the signal strength is decreasing only with the square root of the distance [23]. Even small errors of the received signal strength can lead to big differences of the calculated distance due to the slight curve progression of the RSSI function (Fig. 6). As Fig. 5 shows, the error results in a locale minimum and a maximum of the received signal, resulting from constructive and destructive interference of the electromagnetic wave. In addition, the angle dependence of the tag and the receiving antenna falsify the calculated results.

4.3 Dynamic RSSI Measurement

With regard to the dynamic process of loading and unloading of goods into the container, the tags attached at the goods are passing a few minima and maxima. The resulting curve can be used as the basis for a filtering process which reduces the locale variations of the RSSI measurement.

In Fig. 7 a tag passes the RFID Gate at a minimum distance of 0.61m. As this figure shows, the locale minima and the maxima are only small peaks, which can be easily filtered by a median filter of third order. Filtering of peaks is better with a median filter than with a common mean value because this non-linear filter does not change the characteristic of the curve [24].
In many related works the using of a RSSI map recommended [25], where the local minima and maxima are taken into account for the calculation of distances. To do this, first a test run is used to store the local dependency between the RSSI and the

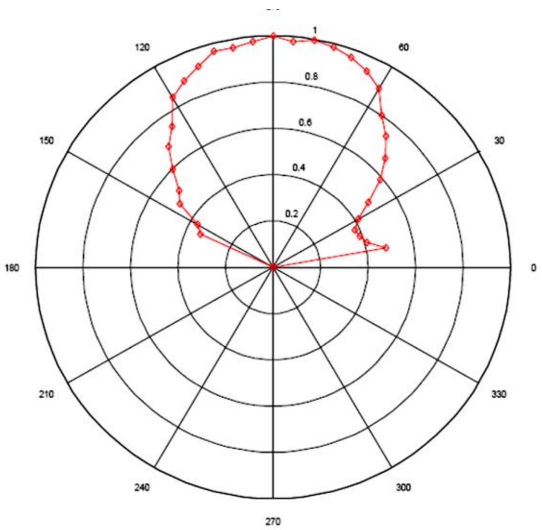

Fig. 8: Antenna pattern of the reader's antenna

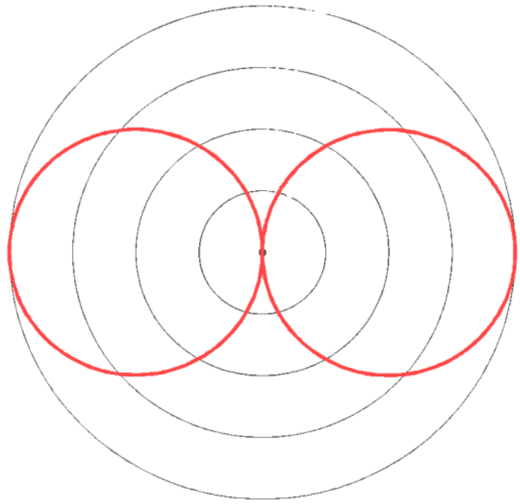

Fig. 9: Dipole like characteristic of a tag

distance. But, because of the dramatically changing environment inside the container a map cannot be drawn. The approach of a dynamic measurement does not need such a map as locale differences of the RSSI values can be filtered.

Using this filter, the calculated minimum distance between the tag and the antenna of Fig. 4 is 0.59m. This result differs only 3.2% from the real value.

4.4 A Cell Based Localization

Because the Readers antenna and the antenna of a tag are not isotropic (Fig. 8), there is a dependency of the received signal between the angle of the Reader´s antenna and the tags. Under this condition, a proper calculation of the distance between the tag and the antenna is impossible. This is why a cell-based localization is preferred, where a calculation of distances is not needed. This method is more stable and absolutely sufficient for the intended purpose.

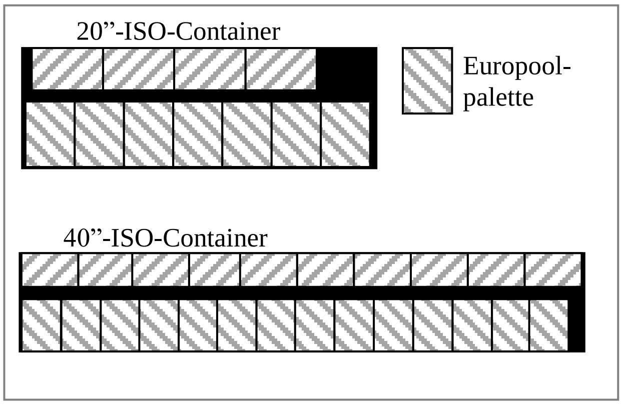

Fig. 10: Arrangements to fill a container with palettes [26]

Each cell is assigned to one of the four Deqtron antennas located at the entrance of the container. Therefore, it is possible to differ between left, right, top, and bottom.
Because the length and the width of a palette can differ, it is necessary to know the orientation of the palette. For the reason that a tag has a dipole like characteristic (Fig. 9), the received signal strength of a tag with a horizontal orientation to the Reader antenna is much higher than a vertical orientated tag. This fact can be used to detect the orientation of the palette.

Fig. 10 shows some arrangements to fill in a container with Europool palettes. As shown in this arrangement, it is absolute necessary to know the left or the right position and the orientation of a palette to extrapolate the location in a container. Now, this knowledge can be used to create context-aware agents or digital waybills in the supply chain.

5 Conclusion

In this work an embedded hardware and a software platform is presented as the basis for an UbiComp environment in the supply chain of perishable goods.

OSGi, with its paradigm of modularization, offers great possibilities to act as a dynamic platform for UbiComp environments. For embedded Systems this framework shows better performance than the complex JADE framework. To realize embedded objects, the whole framework was included on the wireless sensor node Imote2. Running Linux on this sensor node allows installing a JVM, making it possible to install OSGi on this device.

To enable the possibility of context-aware waybills the RSSI at the container entrance is measured with a cheap and a simple method to discern the position of the freights. On the other hand, a single error in localization of the freight in the RFID Gate can lead to a larger error in positioning of the goods.

Although the outcome of this approach is good, an overshoot can lead to erroneously detected tags, which in turn can result a localization error. Cognitive knowledge can be used to decrease this kind of errors. Further research on this topic will be performed to reduce these errors and to acquire reliable localization of the freight.
Nevertheless, these approaches make ubiquitous computing in the supply chain more likely to turn into real application, which enables decreasing the cost and enhancing the quality in logistics.

References:
[1] K.C. Gross: Fresh vegetable quality in the United States: Currentissues. *Acta Hort. (ISHS)*, Vol. 483, 1999, pp. 39-47

[2] R. Jedermann, R, E. De Jong, L. Kleiboer, A. Wessels, S. Yuan, W. Lang, Wireless quality monitoring in the food chain, *6th European Conference on Wireless Sensor Networks, Poster/Demos Session*, No 5432, pp. 232-246

[3] Mehrdad Babazadeh, Reiner Jedermann, Walter Lang, Empirical Issues of a new Environmental Parameters Modeling Technique Using Wireless Sensor Networks, *Proceedings of the 12th WSEAS International Conference on SYSTEMS*, Greece, 2008, pp. 296-301

[4] A. Sklorz, D. Mrugala, W. Lang, IR-Ethylene Concentration Measurement in Fruit Logistics, *Smart Systems Integration 2009*, Brussels, Belgium, 2008, pp. 383-390

[5] R. Jedermann, W. Lang, Mobile JAVA Code for Embedded Transport Monitoring Systems, *Proceedings of the Embedded World Conference 2006*, Nuremberg, Germany, Vol 2, 2006, pp. 771-777

[6] M. Weiser, The Computer for the 21st Century, *Human-computer interaction: toward the year 2000*, 1995, pp. 933-940

[7] B. Schilit, N. Adams, R. Want, Context-aware computing applications, *First International Workshop on Mobile Computing Systems and Applications*, 1994; pp. 85–90

[8] R. Want, B. Schilit, N. Adams, R. Gold, K. Petersen, D. Golberg, J. Ellis, M. Weiser, The ParcTab Ubiquitous Computing Experiment, *Kluwer International Series in Engineering and Computer Science*, Vol. 353, 1996, pp. 45-101

[9] R.H. Clarke, D. Twede, J.R. Tazelaar, K.K. Boyer, Radio frequency identification (RFID) performance: the effect of tag orientation and package contents, *Packaging Technology and Science*, Vol. 19(1), 2006, pp. 45-54

[10] R.C. Dorf, A. Kusiak, Handbook of design, manufacturing, and automation, *Wiley-Interscience*, 1994

[11] F. Brücker, Effizientes Palettenmanagement, *RFID im Blick*, September 2009, pp. 50

[12] H., Jacobi, *Results of the European Pallet Association's EPC/RFID Pilot Program*. EPAL, available at:
http://www.gs1.eu/jscripts/tiny_mce/spirit_docs/upload_0_230/epal_rfid.pdf (12.01.10)

[13] Amani Daknou, Hayfa Zgaya, Slim Hammadi, Herve Hubert, Toward a multi-agent model for the care of patients at the emergency department, *Proceedings of the 10th WSEAS international conference on Mathematical methods*, computational techniques and intelligent systems, 2008, Greece, pp. 264-269

[14] G. Wütherich, N. Hartmann, B. Kolb, M. Lübken, *Die OSGi Service Plattform*. Heidelberg : dpunkt.verlag, 2008

[15] R. Morris, *On the Ciriteria to Be Used in Decomposing Systems into Modules* - Carnegie-Mellon University, 1971.

[16] Moon-Soo Lee, Jong-Woo Choi, Myung-Jin Lee, Kwon Oh-Cheon, Terminal application middleware for the interoperability among telematics services, Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers, Tenerife (Spain), 2005, pp. 139-142

[17] OSGi Alliance, About the OSGi Service Platform, *technical whitepaper, revision 4.1*, June 2007

[18]] F. Bellifemine, G. Rimassa, A. Poggi, JADE - A FIPA-compliant Agent Framework, *Proceedings of the 4th International Conference and Exhibition on The Practical Applicationof Intelligent Agents and Multi-Agents*, London, 1999, pp. 97-108

[19] F. Siebert, *JamaicaVM 3.0 User Manual*. Aicas, Kalsruhe, 2006

[20] H. Chen, F. Perich, T. Finin, A. Joshi, Standard ontology for ubiquitous and pervasive applications, *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services* (MOBIQUITOUS 2004), 2004, pp. 258-267

[21] M. Keskilammi, L. Sydänheimo, P. Salonen, M. Kivikoski, Read Range Analysis of Passive RFID Systems for Manufacturing Control Systems, *Proceedings of the WSEAS International Conferences*, Mexico, 2002, pp. 2081-2085

[22] K. Finkenzeller, *RFID Handbuch*. Hanser, München/Wien, 2006

[23] G. Zanca, F. Zorzi, A. Zanella, M. Zorzi, Experimental comparison of RSSI-based localization algorithms for indoor wireless sensors networks, *European Conference on Computer Systems*, 2008, pp. 1-5

[24] A. Erhardt, *Einführung in die Digitale Bildverarbeitung: Grundlagen, Systeme und Anwendungen*. Vieweg+Teubner, Wiesbaden, 2008

[25] D. Lymberopoulos, Q. Lindsey, A. Savvides, An Empirical Analysis of Radio Signal Strength Variability in IEEE 802.15.4 Networks using Monopole Antennas, *ENALAB Technical Report 050501*, 2005

[26] J. Ihme, *Logistik im Automobilbau: Logistikkomponenten und Logistiksysteme im Fahrzeugbau.* Hanser Fachbuchverlag, München/Wien, 2006, pp. 46-47