Embedded context aware objects for the transport supervision of perishable goods

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: The intelligent freight objects could solve logistical problems themselves to simplify complex logistical planning. The concept of Ubiquitous Computing (UbiComp) presents a model to realize the intelligent objects, which performs local tasks in groups of intelligent objects at their own. The realization of UbiComp needs a platform offering the required intelligence for this task. In order to create context-aware freights in a transport chain, their position in the cargo hold has to be known. This article presents a concept to realize the hardware and software. A system is presented to implement a software representation of the freights to solve logistical tasks. Furthermore, a localization method with RFID is described to create context-aware objects.

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

1 Introduction

In a time where 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. 20% of all perishable food is wasted during transportation according to the U.S. Food and Drug Association (FDA) [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 ([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 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.

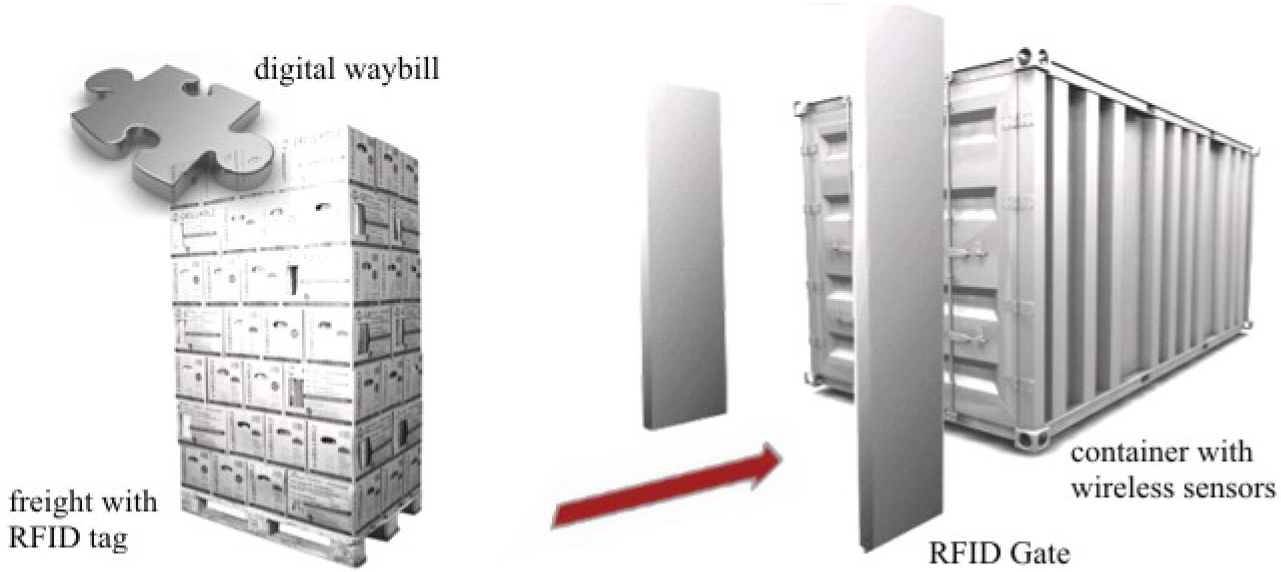

Fig. 1: Concept of the intelligent container

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.

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

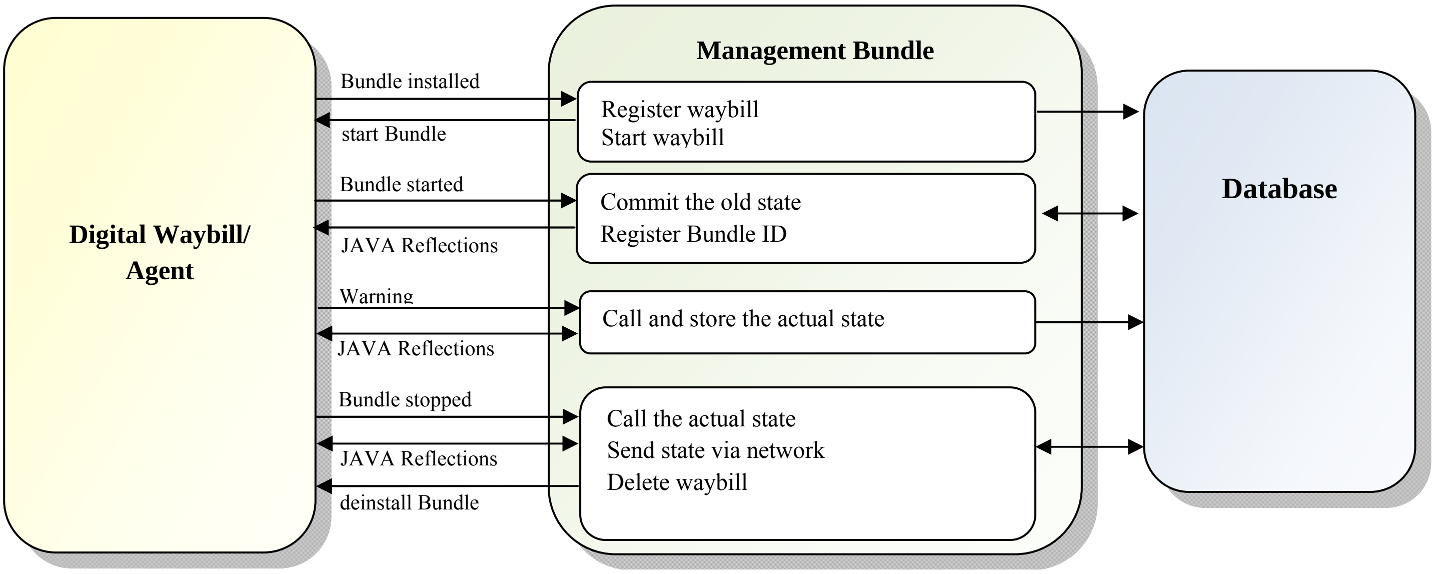

Fig. 2: Implementation of the digital waybill into OSGi

[13]. 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 fixed communication format called FIPA with a size of 1.2 kB. But for an efficient implementation of the waybills this fixed communication format is 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 realization. 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]. 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. Despite this module based system the inter-application communication is near-zero [16]. 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 [17]. 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 is storing the current state of the waybill-bundle into a database allowing seamless transport supervision. JAVA Reflections are used to access all fields and variables of the Bundle for storing them in the database. 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

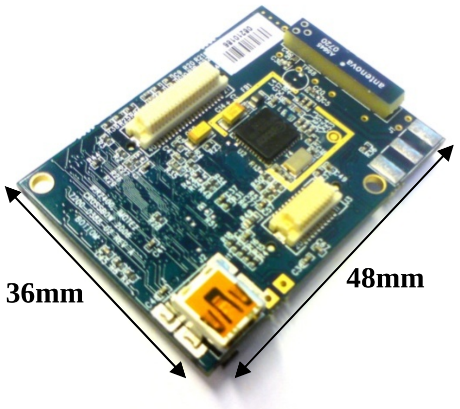

Fig. 3: An Imote2 sensor node

middleware is implemented on the Imote2 (Fig. 3). To run Java on an embeddked 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 ZigBee.

To implement OSGi on the Imote2 as efficient as possible, a small part of the code is compiled. This 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 [18].

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 tend to be highly dynamic and heterogeneous [19].

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 a 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.

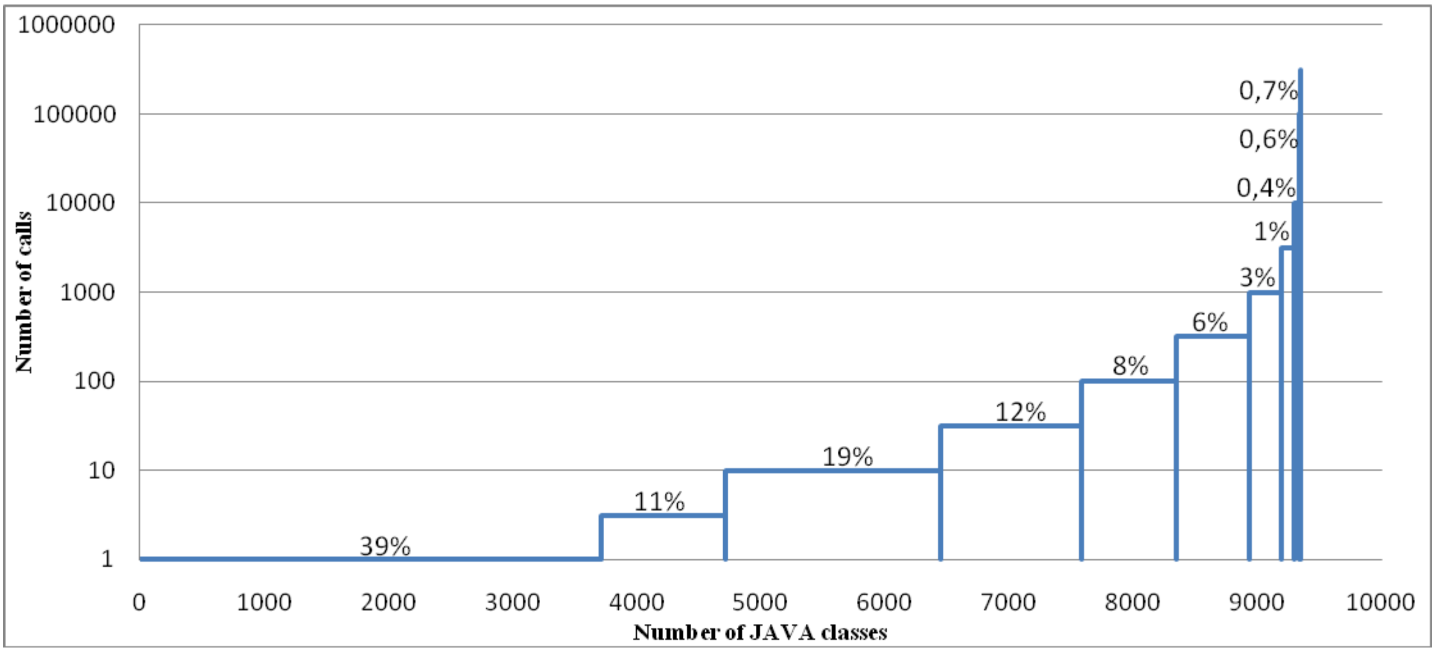

Fig. 4: JAVA method calls during a test-run of OSGi

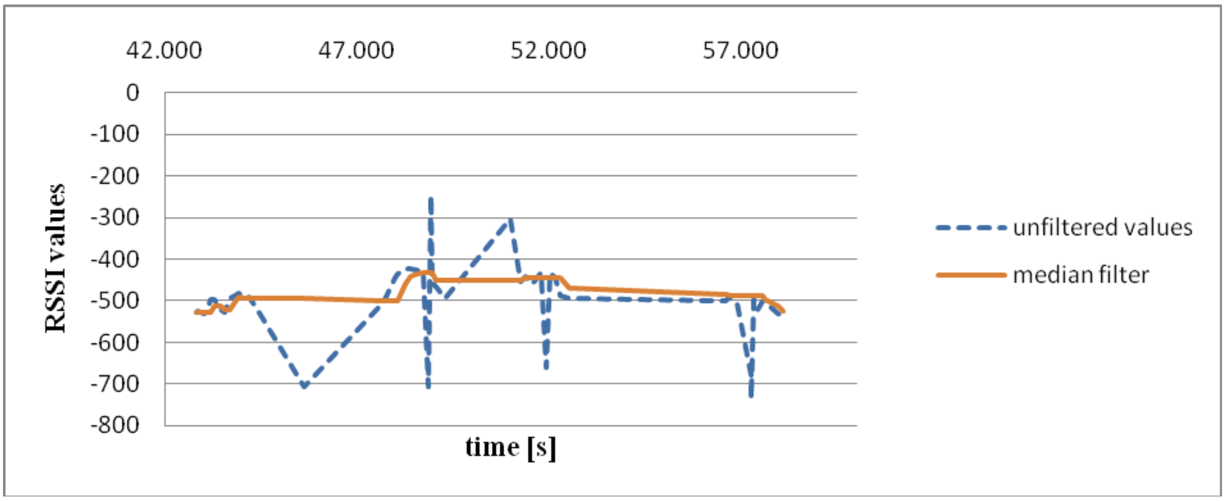

Fig. 2: RSSI values during the passing of the RFID Gate

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. For that purpose, first, the position of the goods at the RFID Gate is detected. Afterwards, it acts on the assumption that the goods will be driven into the container straight ahead. Thus, the horizontal and vertical position of packages must not change during the packing 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.

6.1 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.

4.1 Theoretical Background

For localization in the RFID Gate a long-range RFID system with 868 MHz (Europe) was used. The functionality of this kind of RFID is based on backscattering, where the transponder reflects the electromagnetic wave back to the Reader.

The tag can change its scatter aperture σ in such a way that the information can be transmitted back to the Reader. The power of the reflected wave can be described as follows:

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

From this it follows that the received power at the Reader

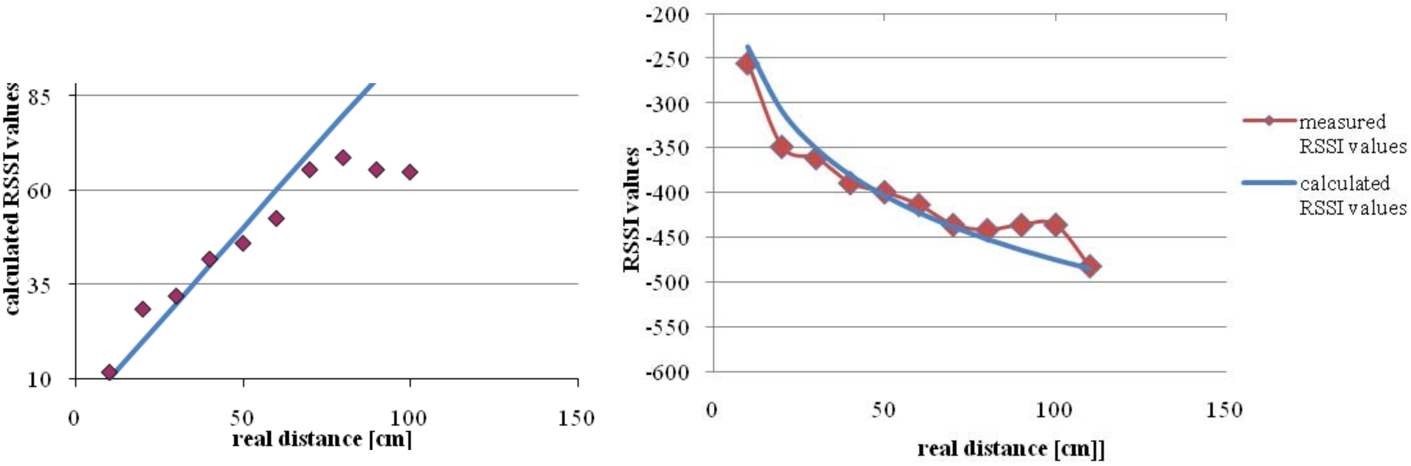

Fig. 6: Dependency of RSSI value and distance

will decrease with the forth root of the distance [20]. 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 [21]. 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. 3 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.

6.2 Dynamic RSSI Measurement

With regard to the dynamic process of loading and unloading of goods into the container, the tags attached in the goods is 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. 5 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 [22].

In many related works the using of a RSSI map recommended [23], 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 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.

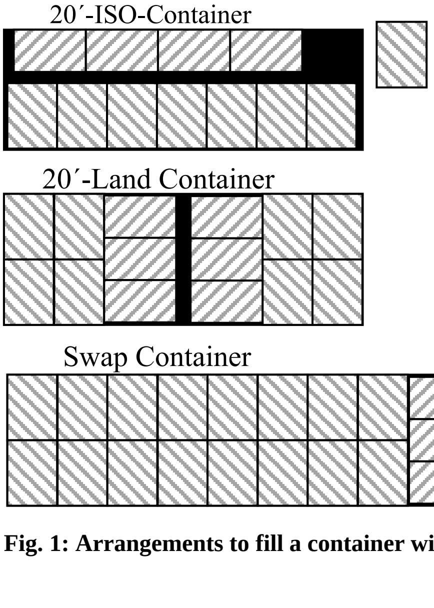

Fig. 1: Arrangements to fill a container with palettes [23]

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.

6.3 A Cell Based Localization

Because the Readers antenna and the antenna of a tag are not isotropic, 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.

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, 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. 7 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.

7 Conclusion

In this work we present the hardware and the software 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 enable the possibility of context-aware waybills the RSSI measurement at the container entrance is a cheap and a simple method to conclude 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. Further research in this topic will decrease the possibility of an overshoot and thus give a reliable localization of the freight.

Nevertheless, these approaches make ubiquitous computing in the supply chain more possible, which enables decreasing the cost and enhancing the quality in logistics.

References:

[1] Gross, K.C.: Fresh vegetable quality in the United States: Currentissues. *Acta Hort. (ISHS),* vol. 483, 1999

[2] Jedermann, R.; Lang, W., The minimum number of sensors - Interpolation of spatial temperature profiles, *Rödig, U.; Sreenan, C.J. (eds.): Wireless Sensor Networks, 6th European Conference*, EWSN 2009.

[3] Jedermann, R.; De Jong, E.; Kleiboer, L.; Wessels, A.; Yuan, S.; Lang, W., Wireless quality monitoring in the food chain, *Pesch, D; Das, S. (eds.): 6th European Conference on Wireless Sensor Networks, Poster/Demos Session*, University College Cork, Cork, Ireland, 2009

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

[5] Jedermann, R, *Autonome Sensorsysteme in der Transport- und Lebensmittellogistik*, Dr. Hut, Bremen, 2009

[6] Weiser, M: The *Computer for the 21st Century*. In: Scientific American, 265, September 1991

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

[8] Want, R.; Schilit, B.; Adams, N.; Gold, R.; Petersen, K.; Golberg D.; Ellis, J.; Weiser, M.: *The ParcTab Ubiquitous Computing Experiment,* Kluwer International Series in Engineering and Computer Science, 1995

[9] Jedermann, R.; Stein, K.; Becker, M.; Lang, W.: *UHF-RFID in the Food Chain - From Identification to Smart Labels*. In Kreyenschmidt. J. (edt.): Coldchain Manangement. 3rd International Workshop, 2.-3. June 2008

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

[11] Brücker, F: *Effizientes Palettenmanagement*. In: RFID im Blick, September 2009

[12] Jacobi, H.: *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] Jedermann R. und Lang W. *Mobile Java Code for Embedded Transport Monitoring Systems*. In: Grote C. und Ester R.: Proceedings of the Embedded World Conference 2006, February 14-16, Nuremberg, Germany. Vol 2., Poing: Franzis Verlag, 2006

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

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

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

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

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

[19] H. Chen, F. Perich, T. Finin, and A. Joshi, Soupa: *Standard ontology for ubiquitous and pervasive applications*. In: The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, MOBIQUITOUS 2004., 2004

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

[21] Zanca, G.; Zorzi, F.; Zanella, A.; Zorzi, M.: *Experimental comparison of RSSI-based localization algorithms for indoor wireless sensors networks*. In: Proc. of ACM RealWSN, Glasgow, Scotland, 2008.

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

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

[23] Ihme, Joachim: *Logistik im Automobilbau: Logistikkomponenten und Logistiksysteme im Fahrzeugbau*. Hanser Fachbuchverlag, München/Wien, 2006