Collaborative Sensing in a Retail Store Using Synchronous Distributed Jam Signalling

Albert Krohn, Tobias Zimmer, Michael Beigl and Christian Decker

Telecooperation Office (TecO), Universität Karlsruhe

Abstract. The retail store environment is a challenging application area for Pervasive Computing technologies. It has demanding base conditions due to the number and complexity of the interdependent processes involved. We present first results of an ongoing study with *dm-drogerie markt*, a large chemist's retailer, that indicate that supporting product monitoring tasks with novel pervasive technology is useful but still needs technical advances. Based on this study, we uncover problems that occur when using identification technology (such as RFID) for product monitoring. The individual identification struggles with data overload and inefficient channel access due to the high number of tags involved. We address these problems with the concept of *Radio Channel Computing*, combining approaches from information theory, such as the *method of types* and *multiple access adder channels*. We realise data pre-processing on the physical layer and significantly improve response time and scalability. With mathematical formulation, simulations and a real world implementation, we evaluate and prove the usefulness of the proposed system.

1 Introduction

Retail stores are a well perceived application area for Pervasive Computing technology. The processes and workflows involved in a retail store span multiple domains, ranging from physical handling of products for logistics, presentation and check-out to information handling, such as required for marketing and product monitoring. Pervasive Computing technology is by nature an ideal means of improving processes in the physical as well as in the informational world, as it follows the idea of pervading the world with interrelationships between physical objects, information and people.

Recommender Systems are a common example to show how ideas of Pervasive Computing technology have been introduced into the retail area. These systems inform the customer about products and current offers in the store using a display or audio output. Some systems even go one step further by personalizing the recommendation to enhance the shopping experience. But generally, most of the technological efforts aim at reducing the personnel costs in the stores. The most cited example is the automatic check-out without a cashier. These self-check-outs are currently under experiment at e.g. Metro's Future Store [1].

To contribute to this complex area with a sustainable novel Pervasive Computing technology, we first conducted a study on the processes and workflows in a retail store. With this study we sought to gain an understanding of the interdependencies of the processes and the real world requirements for workflows in a retail store. It helped us to identify tasks that profit from novel Pervasive Computing technologies and revealed features that such new technology should offer. First results of this ongoing study are presented in this paper. Based on this study we selected one application scenario where we see large gains from introducing pervasive technology – the monitoring of sell-by dates. We also took the existing processes and technologies into account to be able to smoothly integrate the new ideas into existing retail stores.

1.1 Identification and Classification – Tagging on Item-Level

Today, information technology in retail stores is mainly targeted at the *classification* of products. The wide-spread use of EAN (European article numbering system) and UPC (universal product code) barcodes is the reference example for this. Barcodes do generally not identify an individual item but classify it to be e.g. a can of coke. This classification is enough to support tasks in the retail area like ordering or the check-out of customers.

Most of the applications to date that already use the new capabilities of RFID in the retail area only exploit the tagging of groups of products like boxes or pallets, because the individual tagging of products with RFID tags is still too costly, technically not fully matured and not yet completely included in the manufacturing process of packaging of goods etc. Nevertheless, it is widely agreed that RFID has an enormous potential as technological basis for applications in retail stores in the near future.

A drawback that comes along with the *individual* tagging of product is the explosion of data. To take advantage out of the individuality of a single item, an electronic counterpart would have to exist in a data processing system. Handling this mass of data is a complicated task, such that the summarization of local information of single items into groups or classes, in the back end systems, is likely. Further, the potentially huge number of tags communicating in the same radio range severely worsen the problem of access control on the radio channel.

In this paper, we discuss exactly the problems associated with the tagging of products on item level. We see the area of individual identification the most challenging one in the area of RFID technology. The mass of data affecting the radio protocol as well as the processing in data bases is a known issue for the successful future of RFID technology.

In the following, we focus on the application area of product monitoring derived from our ongoing study (section 2). We discuss existing solutions and Related Work in section 3 and summarize the technical requirements in section 3.2. In section 4, we present a novel approach of collaboratively sensing individual information from a large group of goods. Our technical solution is a protocol extension of RFID technology, based on the idea of data pre-processing on the physical layer. We call this approach *Radio Channel Computing*. With the proposed solution we address both mentioned problems of individual tagging: the data explosion and the channel access problem due to the huge number of tags. In section 5, we also present a prototype implementation on wireless nodes and discuss the technical feasibility for the target technology RFID.

2 Processes and Workflows in a Retail Store – A Study

To gain an insight into the workflow of retail stores and their processes, we are conducting a study with one of Europe's leading chemist's companies: *dm-drogerie markt*[1] with more than 1500 chain stores

and 20,500 employees in eight European countries. The products offered by *dm-drogerie markt* are manifold, ranging from toiletries and pharmaceutical products, household articles and pet food, to baby and whole food products. Figure 1 shows a typical view of a shelf in a *dm*-store. The study includes interviews with employees on different hierarchical levels. We conducted interviews with four store managers and were supported by six managers at the headquarters giving background information on the supply chain management and an overview on company wide processes. We visited four *dm*-stores of different size and location to gain a representative view on the workflow, complexity and requirements in this retail store environment. The interviews with the store managers included asking predefined questions about the organization, product monitoring, stocktaking and the internal workflow in the company and stores. These interviews were supplemented with short interviews with some shop employees and on-side demonstrations of the relevant work sequences.

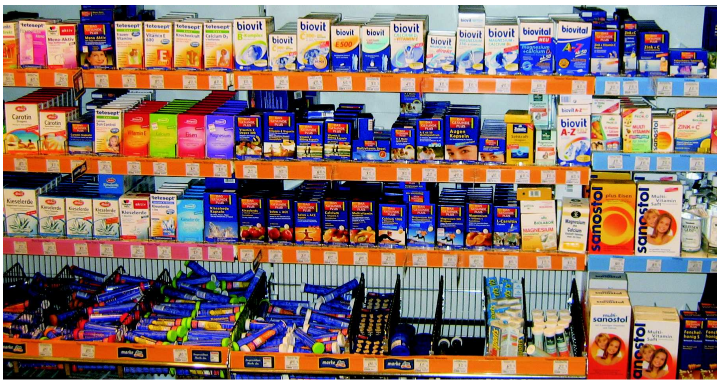

Fig. 1. A typical shelf in a *dm*-store

[1] `http://www.dm-drogeriemarkt.de`

In this paper we focus on handling perishable goods as one facet of the results of this ongoing study that showed very high potential for the application of Pervasive Computing technologies. For chemist's retail stores it is not immediately obvious that monitoring sell-by dates introduces a significant workload. Nevertheless many products, like body lotions and bath additives, in the large assortment of *dm-drogerie markt* carry sell-by dates. To get an idea of how much working time is spent on monitoring, we asked managers of *dm*-stores how product monitoring is organized, how often it is performed and what tasks are involved. In the following section, we present a summary of our study to outline the most important issues in the aspect of monitoring and managing products with sell-by dates.

How often are the shelves checked for perishable products? We learnt that frequent checks that are scheduled on demand by the store manager are in most cases impossible even in small stores with a limited stock. This is due to the number of products with sell-by dates and their distribution throughout the product groups. The headquarter provides predefined schedules for stock maintenance for all *dm*-stores. The checking intervals for product groups range from once every six month to every two months depending on how far the sell-by date is from the production date, how often a product is sold and how large the in-shop stock of a product normally is.

The fixed schedule leads to an unnecessary amount of checks, as in many cases the stock rotation ensures that no products are found that are near their sell-by dates. On the other hand, with fixed check intervals, it is still possible that products that have passed their sell-by dates remain in the shelves until the next scheduled check or a customer discovers them. This could only be avoided if products that reach their sell-by date during the next check interval would generally be removed form the shelves, which is not feasible for products that are only checked every six months. The store managers agreed that it would save a lot of time if the checking could be done on an on-demand basis.

What are the criteria for products to be removed from the shelves? When checking a shelf, the employee removes all products and checks the sell-by dates. Products that reach the sell-by date within the next 8 weeks are removed, reduced in price and put in special sale boxes. Over all product groups that have to be monitored, on average 10 out of 4000 articles are sorted out for sale due to their near expiration. All product monitoring tasks, including continuous stocktaking, are performed during regular opening hours. The store manager schedules the work as short term, depending on the number of customers in the shop and the general workload of the store employees.

How long does it take to check a product group? The checks for sell-by dates are a laborious task. The time needed for checking depends on various factors. Some of them can be influenced by technical aids, some are only connected to the knowledge and experience of the employee performing the check. The product groups in a store can be separated into three classes:

The first class consists only of product groups in which no single product has a sell-by date, such that they require no monitoring.

The second class is made up of the product groups that consist only of products that all have a sell-by date. In this class all products in a product group have to be checked in the monitoring process.

The third class contains product groups in which one can find products with and without sell-by dates. Whenever a product group of this third class is to be checked, the familiarity of the employee with the product group significantly influences the time for the checks. Other important factors on the time necessary

	different products on the shelf	items on the shelf	time to check the shelf	check time per item
whole food products	241	3000	6h	7sec
pharmacological products	487	4000	16h	14sec

Fig. 2. Some numbers on the manual process of checking the goods

are the shape and size of the products and where on the packaging the sell-by date is located. It takes much longer to check small products with tiny printing, like lip sticks, than it takes to check larger products like flour bags. We asked the store managers for the time it takes to check different product groups that significantly differ in their characteristics. In figure 2 we now compare two product groups: *whole food products* (5 shelf segments[2]) and *pharmacological products* (7 shelf segments).

How many items are found? An important indicator when analysing the expenditure of product monitoring is the number of items that are actually identified for removal from the regular stock in individual product groups. In our interviews we learnt that many different factors influence these numbers: The location and size of a store influences what products are sold and how many times. To some extent this determines also how exactly the assortment in a given store is made up. Large stores in shopping areas and malls with big parking lots attract more customers that buy bulky and heavy goods. The customers of theses shops tend to be younger than in other areas which is reflected by a higher sales volume in whole food, beauty products and baby products. Stores where the share of elderly people among the customers is high, tend to sell more pharmacological products. Additionally, some products like sunscreen are

[2] a shelf segment is 1 m wide and 1,80 m high

subject to strong seasonal fluctuations. So it can be safely assumed that no two of the 1500 *dm*-stores offer exactly the same assortment.

It is not feasible to provide each store with an individually optimised schedule for product monitoring. So a general schedule has to be very defensive to make sure that it is applicable for all of the stores. The large variety of assortments and the differences in the turnover in the product groups lead to very diverse numbers of items that have to be removed due to expiration. According to the store managers, in around 4000 single items, the number of products with critical sell-by dates can be none, some or up to 40 depending on the store and the checked product group. On average, only 10 out of 4000 items have to be sorted out during a check, while checking of these items takes more than 12 hours.

2.1 Implications and Assumptions – Check on Schedule vs. Check on Demand

The limitations of efficiency using a fixed, centrally prepared monitoring schedule have been discussed in the previous section. As mentioned, schedule-based checking leads to an unnecessary amount of checks introducing high costs due to the expenditure of time for the checks. An alternative to *checking on schedule* would be *checking on demand*, which is more flexible with respect to the individual product group turnaround times as well as to the different store profiles. Checking on demand would improve the intervals between individual checks saving personnel costs. In this context the granularity of the partitioning of products for the checks should be decreased below product group level. Checking individual shelves instead of whole product groups (up to 10 shelf segments) would add to the flexibility and reduce the items involved in one single check down to around 400, based on the figures derived from our survey. For shelves that need higher attention, the check rate could be increased with still lower over-all costs than with using fixed schedules on product group level. Changing the task of sell-by dates checking from *checking on schedule* to *checking on demand* would positively influence the overall efficiency of the workflow in a *dm*-store. Based on this idea, we envision a technical system that allows to change the scheduling of product monitoring by introducing Pervasive Computing technology into the workflow.

Further, as a second result of our study, the store managers are very interested to monitor the changes over time of the sell-by dates of a certain product group or all products of one shelf. For example, consider that after new items have been placed on a shelf a continuous process of selling and reloading of the shelf follows. During that continuous process the products become more and more mixed concerning their sell-by dates, due to activities such as customers taking products from the back of the shelf, personnel placing "fresher" products in front of older ones, and other random re-ordering of the shelf space. The monitoring of the development of the sell-by dates of a defined group of products is a helpful instrument for ordering, advertising and for the optimisation of the range of products.

To enable both tasks – the *check on demand* and the *generation of the overview of sell-by dates* – we propose an extension to existing RFID protocols. The technical details are discussed in the following sections.

3 Collaborative Sensing for Group Profiling

For the previously mentioned tasks of checking the sell-by dates and the generation of the overview of the sell-by dates, we want to support the employees by providing a *profile* of the sell-by dates of a given group of items. A *profile* is a histogram showing the sell-by dates of all observed goods.

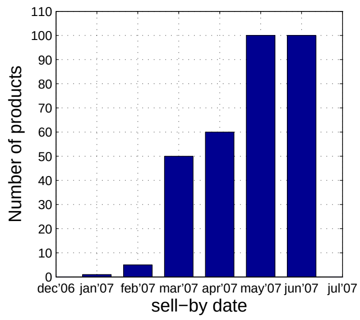

Fig. 3. Profile of sell-by dates of products

Fig. 4. Ordering products

Figure 3 shows such a profile. The profile does not give information on the individual identification of the products but only gives the number of products with the same sell-by date. For the generation of the desired profiles, the identification is not necessary. The resolution and range of the time axis is a matter of implementation. With this profile, the employee can decide whether a manual check is necessary and can at the same time monitor the development of the sell-by dates. If e.g. all observed goods expire very far in the future, a manual check would obviously not be necessary. Such a profile provides the basis for the on-demand checks mentioned in section 2.1.

The technology that we propose will generate these profiles by *collaboratively sensing* the sell-by date among selected products. It is based on RFID technology. We therefore assume, that all products are tagged with RFID transponders and carry their sell-by date in the tags. To avoid time-consuming additional tasks, we envision combining the process of sell-by date profiling with the inevitable ordering process. When ordering new products, the employee scans a product identification – normally a barcode – on the shelf to trigger the process in a back-end system (see figure 4). While pointing at the barcode to scan, we could evoke a collaborative sensing of the sell-by dates of selected products that are in radio range and generate a profile of their sell-by dates. The radio range for HF/UHF (typically around 1m) would cover 200-300 items. The profiles of sell-by dates could be displayed on the scanner or collected in the back-end for generating a store wide profile.

To seamlessly include the generation of profiles into the ordering process and avoid additional inconvenience, the primary factor is the time needed. The scanning of a barcode for the ordering process would typically need a fraction of a second. Therefore, the included sell-by dates scanning process must be performed in a similar time. If it takes too long, it is not feasible. The task should not produce additional inconvenience and should appear to be real-time – typically faster than 50ms.

3.1 RFID Standards, Prevailing Protocols and Related Work

Smart Shelves The tracking or sensing of products in stores is an intense discussed topic. One suggested solution is the use of so called *smart shelves*. These shelves have multiple RFID readers embedded and are therefore able to continuously scan all the products placed in them. In [2] even basic interaction of customers with the products can be monitored. Nevertheless, installing an infrastructure, which completely covers all product areas and stocks in a store is a complex and costly task. One out of many problems is that of antenna interference when many readers are installed in a dense setting. Additionally, the readers have to be connected to a power supply and in some cases to a data processing system. This disturbs, for example, frequently changing installations for special offers and needs frequent reorganization of the whole system. While a self-organizing systems approach may be considered as a solution, they increase the complexity.

Active Tags Several companies have specialized in smart product monitoring in the retail area, with perishable goods being a popular focus for technical solutions. An example where the sell-by date is monitored in a system based on smart tags is marketed by Infratab [3]. Infratab's FreshAlert^{TM} tags can monitor the shelf life of perishable goods. The technology is based on active tag technology including a temperature sensor, an indication light and a battery. Infratab proposes a target price of 1$ [4] which we consider to be too high for our target application.

Radio Frequency and Read Range RFID technology is generally classified by its target operation frequency. The demand for longer read ranges and higher data rates drove the communication frequencies higher and higher. To date, the most common standards in the HF- Band (13 Mhz) are the ISO 15693 [5] and the ISO 18000-3 Mode 1 and 2 [6]. In the UHF band (868/916 MHz) the most important efforts are the ISO18000-6 and the standard contribution from EPC global [7]. For the target application identified in our study, we are especially interested in high data- and identification rates as well as in read ranges in the area of around one meter. Even though a read range of one meter is hardly possible with available HF-band technology, both HF and UHF are considered as state of the art for the target application.

Identification Speed According to the standards, the fastest identification processes in the HF-band can be achieved with the ISO 18000-3 Mode 2, reaching up to 200 items/s. Other protocol improvements have been achieved and have been contributed to the Mode 5 specification, which is now discontinued due to the high requirements on the hardware. Using this discontinued mode 5, in [8], the authors calculate that an identification of 500 tags including 100 Bytes of data read from each tag should be possible in around 800ms. Nevertheless, both results only refer to an idealized theoretical calculation and give no statistical confidence. According to an article in the RFID journal in June 2003 [9], the latest technology from Magellan Technology Pty Ltd [10] can identify 1200 tags per second using the ISO 18000-3 Mode 2 when tags are sequentially running through a tunnel reader at high speed. This does not mean that 1200 tags can be read in one second in a static setting. Magellan has instead shown another demonstration, where they identify 100 tags within a second in a nearly static setting. The numbers given for identification and read processes have always to be read very carefully to not be misinterpreted. As an example for a realistic estimation of the speed of identification processes, we refer to the work of Vogt in [11]. There an identification scheme (based on slotted Aloha) is statistically modelled under realistic assumption for collisions and optimised for a fast identification of many tags in a static setting. According to his work, the optimal number of slots and cycles for a 99% detection confidence of 200 RFID tags would be higher than $256 * 13$ slots, whereas each slot carries a complete packet. This theory of the identification scheme is generally not limited to a certain target technology (HF or UHF). As a theoretical best-case calculation, we pick values from the most actual physical layers of HF/UHF standards. There, the shortest signals from a tag are $6.25\mu s$ (for UHF) and $14.2\mu s$ (for HF, ISO18000-3 Mode 2). Assuming 64 Bit identifiers, these physical layer parameters and 200 devices, the whole process would still result in more than 3000ms not considering the overhead through messages headers and packets from the reader which are significant.

Identification Protocols The identification process in the presence of many tags has been discussed widely. A typical approach is the so call *binary search* method. The reader sends out a mask of Bits, the tags compare it with their ID, answer and then the reader sends out the next mask trying to singulate one RFID tag in the read range. This method is explained in detail in [12]. The new standards in the UHF band include mechanisms to speed up the identification process. One approach is e.g. to use a 16 Bit random number for the separation of RFID-tags instead of the complete ID that carries 96 Bit or more. Figure 5 shows such a mechanism. The reader first selects the target group with e.g. filters

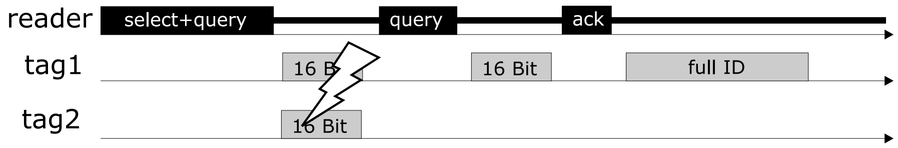

Fig. 5. Using a 16 Bit random number during Identification process

or masks and then sends a "query" command. This command includes as well how many slots follow from which the tags can pick one randomly. Afterwards, the tags pick a random 16 Bit number and send it in the according random slot. If the reader detects a collision, the process is repeated. If the reader can receive a valid 16 Bit number, it then replies and polls the whole ID and then reads or writes data, deactivates the tag etc. As a best case assumption, the protocol of query, reply, ack and ID-reply for a single tag would take (assuming again to date EPC/ISO physical layers) around $2100\mu s$. The random selection is a slotted aloha process know to have a maximum throughput at around 36%. Assuming this and 200 tags, we get something around 1.2 seconds as the process time for reading the IDs of 200 tags. This is of course a very rough rule of thumb calculation but gives a first idea of the timings. This calculation does not include reading or writing of stored data in the tags. It's only the identification!

3.2 Problem Statement and System Requirements

Looking at the examples of the previous section and the target process time of below 50ms, there is a mismatch in technology. As a conclusion of the above analysis of state-of-the-art RFID technology we do not promote a standard protocol for our use case. Smart shelves are too costly, inflexible and complex, while standard RFID tags with mobile readers are not fast enough. To support our target scenario, we propose an extension of existing RFID technology to support the user in her activities rather than completely redesigning the whole workflow which we identified to be too complex and too expensive. We envision a system of *low complexity* and *low hardware requirements* similar to current passive RFID technology where a high number of tags can be efficiently read out in real time. Such a system should also have a *fast and constant response time* (typically faster than 50ms) – even for a large number (1000) of items and should *not need any additional technology or infrastructure*. The constant and predictable response time is a very important point especially for the inclusion of such processes in the normal workflow. With a known reply time, the equipment can work with known delays and the operators do not have to consider the side conditions of the actual task but follow a given scheme without any adaptation to the situation. This is also a big step towards the automation of such a process.

4 Radio Channel Computing

To technically realize a system fulfilling the requirements summarized in section 3.2, we propose a radical new technical approach that we call *Radio Channel Computing*. The idea is, to take advantage of the characteristics of the radio channel itself to perform elementary computation with transmitted data during the transmission on the channel. With this method we can – for the use case discussed in this paper – achieve a *constant, ultra fast* reply time *independent* of the number of tags actually involved in the process.

4.1 Related Work For Radio Channel Computing

The most important ingredients for realizing Radio Channel Computing include early and fundamental ideas of information theory and reach back until the 1940's. The major parts are:

The method of types In [13], the basic of Shannon's discoveries and formulation of information theory ([14]) is presented in the context of applications. Roughly spoken, the idea of the method of types is to compress a vector of measurements by partitioning it into subclasses. All elements of the vector are then assigned to an according subclass and only the number of elements in a subclass in then transmitted.

Superimposed codes In [15] the authors derivate a method to detect codes that have been superimposed on the channel. This idea also dates back to a very early work of Kautz and Singleton ([16]) in which they state: "A binary superimposed code consists of a set of code words whose digit-by-digit Boolean sums $(1 + 1 = 1)$". This idea will be taken over and applied to the area of radio communication with RFID.

Binary adder channels A fundamental work on the use of superimposed analogue signals for a multi-user binary adder channel can be found in [17]. The authors discuss physical constraints of super positioning of non-orthogonal signals on a channel. The combination of the method of types and multi user adder channels has also been proven to be optimal efficient for distributed detection in [18].

Theory of small samples It is a common approach in information theory to prove a concept by it asymptotical behaviour. For our case, we are more interested in results that refer to small samples. We apply standard estimation theory and confidence intervals here.

4.2 Multi-SDJS

Multi-SDJS is an extension to Synchronous Distributed Jam Signalling presented in [19]. It combines the above mentioned ideas in a system, that superimposes jam signals on the radio channel. In the context of this paper, the radio channel is supposed to be a binary OR-channel. Two or more simultaneous transmitted signals result in only one in the receiver. In contrast to [17], the receivers are not able to analyse the details (e.g. number) of super positioned signals but only detect the existence of them. This makes the implementation of the system much easier. To achieve the encoding of the sell-by dates into the SDJS scheme, we apply the *method of types* by sub classing the sell-by dates in e.g. units of months. Subsequently, for each subclass representing a certain time interval, the number of goods carrying the according sell-by date are transmitted. The transmission of each of these numbers is done applying the ideas of *binary adder channels* and the *theory of small samples* through the use of the previously developed SDJS theory [19]. Generally, a SDJS scheme is a number of time slots, where devices with

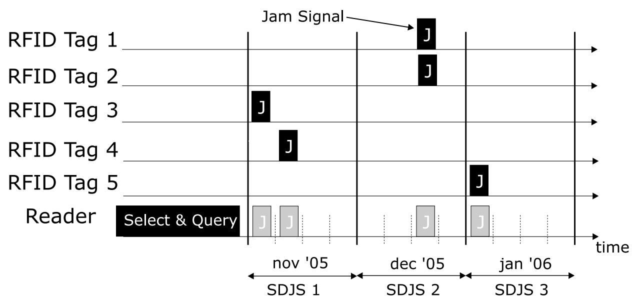

Fig. 6. A Multi-SDJS scheme with three concatenated SDJS schemes

radio communication – in our case RFID tags – emit jam signals into randomly selected positions. Using the same principles as the original SDJS [19], the jam signals carry neither a signature nor an identification. The *presence* of signals only inform about the *existence* of products with the according sell-by date. The occurrence of collisions is included in the SDJS theory [19]. For the use-case of this paper, SDJS schemes are used to estimate the number of goods carrying a certain sell-by date. Multiples of these schemes are concatenated to span the entire time-span of sell-by dates. After processing the multi-SDJS scheme, the receiver has an estimated number of goods for each subclass of dates – the desired profile can be generated. Figure 6 shows how the Multi-SDJS scheme works. The Multi SDJS-scheme starts with the "select & query" command of the reader containing the filter and masks for the selection of tags. A number of SDJS schemes then follows (in this case three), each representing a certain sell-by date. Each of these sub-SDJS schemes contains a number of slots for the positioning of jam signals. During a single SDJS scheme, e.g. during the scheme for dec '05 in figure 6, the tags randomly select a slot and send a jam signal, when their sell-by date is dec '05. The reader collects theses jam signals and uses them to estimate the number of tags for that date.

4.3 New Aspects for the Estimators of SDJS

For our target application, the SDJS theory of [19] requires significant extension. In this section, we will present two new important aspect for the use of SDJS in this retail scenario. First of all, the maximum likelihood estimator (MLE) will be once more discussed for our target setting. As a second step, we will present a maximum a-posteriori (MAP) estimator that enables us to give confidence intervals for the estimations. The following discussion refers to single SDJS schemes and therefore addresses the estimation of number of products that carry the same sell-by date and participate in the same single SDJS scheme.

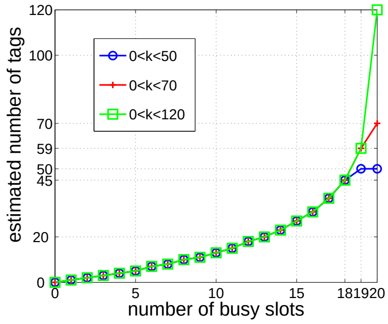

Fig. 7. Invariance of MLE for $s = 20$ and $k_{max} = 50, 70, 120$

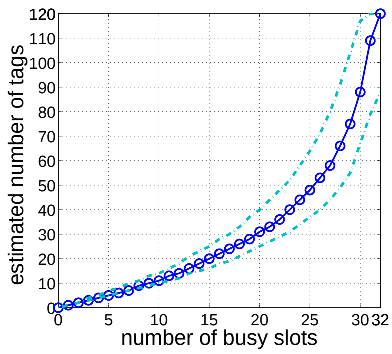

Fig. 8. MAP for $s = 32$ and $k \in [0, 120], \beta = 0.9$

Invariance of the MLE The maximum likelihood estimator of SDJS has the advantage, that the probability distribution $P_k(k)$ of the number of tags k carrying the same sell-by date does not need to be known. Nevertheless, at least the interval $[0, k_{max}]$ of how many tags are possibly there needs to be known. Applying SDJS to the target application, it seems impossible to even only roughly predict the quantity of goods with a certain sell-by date. Fortunately, also this constrain can be loosened when taking a closer look on the ML point estimator. The MLE has an invariance under certain conditions, shown in figure 7. The example is based on a SDJS scheme with $s = 20$ slots. The ML point estimators are given as the three curves. It can be noticed that for all number of received jam signals $a \leq 18$, all three estimators result in the same values even though the preconditions (meaning the range of the number of items) are different. This is due to the nature of the estimator and the underlying process. The ML-estimator ($\arg\max_k\ P_{a|k}(a|k)$) only depends on the mapping that SDJS realises. Applying the additional side constraint of an estimation range $[0, k_{max}]$, an estimation $\tilde{k}$ that reaches out of the range $[0, k_{max}]$ has to be corrected to a lower value $\tilde{k} \leq k_{max}$. In figure 7, for $a > 18$, the estimators give different values for the different preconditions $k_{max} = 50, 70, 120$. It is clear that e.g. for the case $k \in [0, k_{max} = 50]$, the estimator can never result in a higher estimation value than 50. Therefore, the estimator curve is limited to $\tilde{k} = 50$. For $s = 20$ and $k_{max} > 59$, the ML-estimations for $0 < a < 20$ will stay invariant of the actual k_{max}. Only for $a = s$ the estimation is always $\tilde{k} = k_{max}$.

Therefore, we can use only one lookup table for different settings as long as we can assume that the upper bound k_{max} of the possible number of devices is above a certain value. This threshold is *only* given through the number of slots s. For our example ($s = 20$), this threshold is $k_{max} = 59$.

MAP Estimation In the case of a known a-priori distribution, we can improve our estimation by including this knowledge into the estimator. We do this by using a *maximum a-posteriori* estimation (MAP).

$$\begin{aligned} \tilde{k}_{MAP} &= \arg\max\nolimits_k \ P_{k|a}(k|a) = \arg\max_k \ \frac{P_{a|k}(a|k) \cdot P_k(k)}{P_a(a)} \\ &= \arg\max\nolimits_k \ P_{a|k}(a|k) \cdot P_k(k) \end{aligned} \tag{1}$$

For each estimation, the probability $P_a(a)$ is identical and can therefore be left out in (1). As mentioned above, it seems impossible to give those distributions of number of goods $P_k(k)$ with equal sell-by dates as an information input to the estimator. For simplicity, we assume an even a-priori distribution ($P_k(k) = const$). Any other more realistic distribution would of course improve the over all result. With this even a-priori distribution we get the same point estimation results as for the ML estimation, as the constant term $P_k(k)$ does not change the maximum of (1). Yet we can now also give confidence intervals for our estimation. We define a *trust level* β (typically $\beta = 0.9$) and can then give a confidence interval $[\tilde{k}_{min}, \tilde{k}_{max}]$ for our estimation with the certainty of β. Figure 8 shows an MAP estimation for SDJS. The dotted lines are the lower and upper limits of the $\beta = 0.9$ confidence interval. This means that e.g. for $a = 15$, the MAP point estimation would give $\tilde{k}_{MAP} = 20$ and statistically, 90% of all possible cases that result in $a = 15$ originate from the interval $\tilde{k} \in [16, 25]$.

4.4 Accuracy Evaluation and Properties SDJS

Primarily, the intention of Multi-SDJS was to create a fast estimation not focused on accuracy but more on speed and scalability. Normally, wireless packet oriented protocols like Zigbee, Bluetooth, RFID, WLAN etc. require more time when the number of information to be transported over the same bandwidth rises. In contrast to this, the speed of SDJS does only depend on the number of slots and not on the number of tags sending. The only draw-back that comes with more tags sending is the decrease in accuracy. This can be seen in figure 8. The confidence interval increases when increasing the number of items. However, for the application scenario, this is exactly what we wanted: a precise estimation for small numbers and a rough idea for the larger groups.

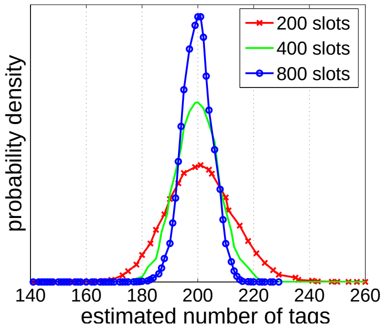

Fig. 9. Distribution of the estimation results for $k = 200$, $s = 200, 400, 800$

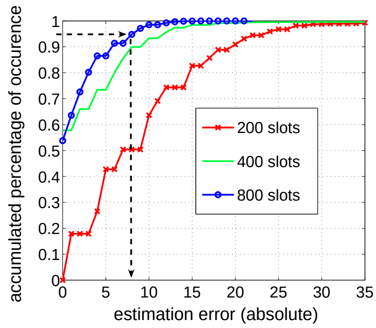

Fig. 10. Error distribution for the estimation of k=200 tags

There are different options to increase the accuracy of SDJS. The most obvious possibility is to do multiple SDJS schemes (for the same sell-by date) and combine the measurements in a joint probability distribution $P_{\underline{a},k}(\underline{a}, k)$ and use again an ML or MAP estimation. As the deeper discussion of estimators is not the focus of this paper, we want only to give some idea of the accuracy that can be achieved by using the presented SDJS estimators from section 4.3. We use a MAP estimator and vary the slot numbers s to show how that influences the estimation quality. We are interested in how accurate we can estimate 200 devices. In figure 9, three curves for the estimation of $k = 200$ devices are shown. They represent the distribution of the estimation results for $s = 200, 400, 800$ slots. Even though the estimation results would result in discrete values, we smoothed the graphic with a spline for a better graphical representation. In figure 10, the accumulated error distribution is shown. This figure is based on the real discrete error probabilities of the discrete estimator tables; it therefore carries the steps of the discrete values of the estimators. To read out from figure 10 how big the error (with e.g. a confidence of 95%) for an estimation of 200 devices based on 800 slots is, we have to start at ordinate value 0.95 (see arrow) and then read out at the abscissa, that all errors will be smaller than 8. That means, that for an experiment with 200 tags and 800 slots we expect 95% of the experiments to have an estimation result error of maximum 7 tags.

5 Implementation and Technical Evaluation

To prove the concept of Multi-SDJS we fully implemented SDJS on our wireless sensor platform: the Particle Computers [20]. In our setting (see figure 11), the Particle Computers are configured to behave like passive tags: A “reader” sends out a start signal and then all devices contribute to the Multi-SDJS signalling. For our scenario of monitoring sell-by dates, we use a Multi-SDJS scheme similar to the one explained in figure 6. As a demonstration example, we took a random distribution of sell-by dates and implemented the dates on the Particle Computers. We then performed the Multi-SDJS scheme. The "select & query" command took approx. $900\mu s$, the slot-time was $64\mu s$. In figure 12 the profiles of the real dates and the estimated ones are compared against each other. In this example, we can benefit from the typical characteristic of SDJS that small numbers are estimated better. The three products that expire in February are estimated more precisely than the higher number for June. As we are especially interested in products with critical sell-by dates, which are typically only a few, we get a very good estimation for them.

Fig. 11. Prototype implementation of Multi-SDJS on Particle Computers

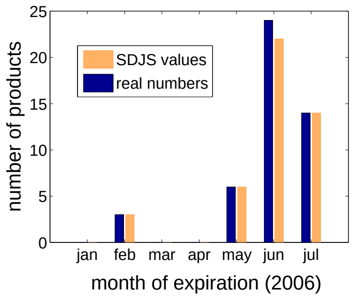

Fig. 12. Profile of sell-by dates of 50 products

Future implementations of Multi-SDJS can be realized on RFID hardware platforms, since the operational demands do not exceed those of standard RFID technology. As a speed comparison, we take the same numbers of the best case calculation from section 3.1 and use them now as well for the Multi-SDJS scheme. Assuming a sell-by date time resolution of one month, a total time coverage of 4 years, 100 slots per date, again around $10\mu s$ for a slot and 1ms for the "select & query" command, the Multi-SDJS scheme would take around 49ms – independent of the number of tags present. The speed is within the cognitive bounds of human, real-time perception and even out-performs the speed requirements give in section 3. As Channel Computing with Multi-SDJS is a very specialized solution for the generation of the mentioned sell-by date profiles, the speed comparison to standard RFID protocols is somewhat improper, given that RFID protocols are not optimized for this particular task and transport unnecessary information. Nevertheless, in the calculation example, Multi-SDJS speeds up the generation of such profiles by 6000% without requiring any additional technological advances. Turnaround times in the tags are not critical as the tags do only one switch to transmission after the query command of the reader. During the slots of the Multi-SDJS scheme, the tags are in transmission state and just pick a slot to transmit their jam signal. The discussion of the best waveforms of the jam signals is not in the focus of this paper. Nevertheless, in the prototype implementation it has already been proven that jam signals can be detected without carrying a packet frame format or preamble or any other signature. Their length was mainly restricted by the sampling speed of A/D conversion and was comparable of the transmission time of 6 Bit of the underlying physical layer. Therefore, the assumption for a SDJS slot time to be in the area of a single RFID Bit-time is in fact realistic. The clock drifts are not critical either. The SDJS-scheme can easily handle a 10% shift of the jam signals. For the given example, this would mean a 10ppm clock, which is a rather high requirement to RFID hardware. Additional (in our case five extra) resynchronisation packets from the reader can easily relax this requirement to 100ppm or less.

6 Conclusion

The ongoing study with *dm-drogerie markt* we presented in this paper provides basic input to our research on novel Pervasive Computing technologies for the retail area. On basis of this study we were able to identify various interesting and promising applications of which we focused on manual monitoring of perishable products by checking sell-by dates. The interviews with the store managers revealed that seamless integration of new technologies and workflows will be essential for their success. We are convinced that in order to assure the real-world applicability of novel Pervasive Computing technology, maintaining close contact with its potential users is necessary.

RFID is widely agreed to be the upcoming technology for logistics and all kinds of workflow and supply chain management in the retail area. Thus it clearly has the potential to be among the novel computing technologies that will pervade our lives. We reviewed state of the art RFID technology and analysed its momentary weaknesses for the deployment in a challenging new application area. This resulted in the proposal of *Multi-SDJS*, an enhancement of existing RFID protocols. We improved response time and achieved constant and predictable scalability and realized a data pre-processing on the radio channel. The underlying idea of *Radio Channel Computing* is the basis for our ongoing research in this area. Many other research areas like data representation and combination, cooperative transmission and distributed estimation flow into this topic. We believe the potential of Radio Channel Computing and also Multi-SDJS in particular goes far beyond the focus of this paper. Being able to sense information in an ad-hoc manner from a large population using a simple and well scaling protocol opens a vast field for new applications like e.g. real time sensor fusion in general. The constant reply time of SDJS is also an ideal pre-requisite for its use in fully automated processes.

7 Acknowledgements

The work presented in this paper was partially funded by the European Community through the project *CoBIs* (Collaborative Business Items) under contract no. 4270 and by the Ministry of Economic Affairs of the Netherlands through the BSIK project *Smart Surroundings* under contract no. 03060.

References

1. Metro group - future store initiative. Website, accessed: 10/2004. http://www.future-store.org.
2. Christian Decker, Uwe Kubach, and Michael Beigl. Revealing the retail black box by interaction sensing. In *Proceedings of the ICDCS /IWSAWC*, Providence, USA, 2003.
3. Infratab Inc. Website, accessed: 10/2004. http://www.infratab.com/.
4. RFID Journal. Rfid tags for monitoring shelf life. Website, accessed 10/2004. http://www.rfidjournal.com/article/articleview/428/1/1/.
5. ISO/IEC/JTC 1/SC 17. Identification cards - contactless integrated circuit(s) cards - vicinity cards. 2001.
6. ISO/IEC/JTC 1/ SC 31/WG 4. Automatic identification - radio frequency identification for item management communications and interfaces part 3: Physical layer, anti collision system and protocol values at 13.56 mhz mode 2. 2001.
7. Epcglobal Inc. Website, accessed: 9/2004. http://www.epcglobalinc.org/.
8. Graham Murdoch and Tim Frost. A comparison of the iso15693 and iso18000-3 mode 5 magellan protocol. 2001.
9. RFID Journal. Magellan unveils tunnel reader. Website, accessed 9/2004. http://www.rfidjournal.com/article/articleview/445/1/1/.
10. Magellan Technology Pty. Ltd. Website, accessed: 9/2004. http://www.magtech.com.au/.
11. Harald Vogt. Multiple object identification with passive rfid tags. In *IEEE International Conference on Systems, Man and Cybernetics (SMC '02)*, October 2002.
12. Klaus Finkenzeller. *RFID-Handbuch*. Carl Hanser Verlag München Wien, 2000.
13. Imre Csiszár. The method of types. *IEEE Transactions on Informatio theory*, 44(6), 1988.
14. C. E. Shannon and W. Weaver. The mathematical theory of communication. *University of Illinois Press*, 1949.
15. P. de Laval and S. Abdul-Jabbar. Decoding of superimposed codes in multiaccess communication. In *8th European Conference on Area Communication, EUROCON*, 1988.
16. W. H. Kautz and R. C. Singleton. Nonrandom binary superimposed codes. *IEEE Transaction on Information theory*, 1964.
17. I.Bar-David, E.Plotnik, and R.Rom. Limitations of the capacity of the m-user binary adder channel due to physical considerations. *IEEE Transaction on Information Theory*, 40(3):662–673, 1994.
18. K. Liu and A. Sayeed. Optimal distributed detection strategies for wireless sensor networks. In *42nd Annual Allerton Conference on Communications, Control and Computing*, 2004.
19. Albert Krohn, Michael Beigl, and Sabin Wendhack. SDJS: Efficient statistics for wireless networks. In *Proceedings of the 12th IEEE International Conference on Network Protocols*, Berlin, Germany, 2004.
20. C. Decker, A. Krohn, M. Beigl, and T. Zimmer. The Particle Computer System. In *IPSN Track on Sensor Platform, Tools and Design Methods for Networked Embedded Systems (SPOTS). Proceedings of the ACM/IEEE Fourth International Conference on Information Processing in Sensor Networks*, Los Angeles, USA, 2005. to appear.