ECIS 2008 Proceedings

European Conference on Information Systems (ECIS)

2008

Design Guidelines for RFID-Based Applications in Manufacturing

Lenka Ivantysynova
Humboldt Universität zu Berlin, lenka@wiwi.hu-berlin.de

Holger Ziekow
Humboldt-Universität zu Berlin, ziekow@wiwi.hu-berlin.de

Recommended Citation

Ivantysynova, Lenka and Ziekow, Holger, "Design Guidelines for RFID-Based Applications in Manufacturing" (2008). *ECIS 2008 Proceedings*. 206.
http://aisel.aisnet.org/ecis2008/206

DESIGN GUIDELINES FOR RFID-BASED APPLICATIONS IN MANUFACTURING

Ivantysynova, Lenka, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, D, Lenka@wiwi.hu-berlin.de

Ziekow, Holger, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, D, Ziekow@wiwi.hu-berlin.de

Abstract

The RFID technology opens new opportunities to monitor and control manufacturing processes. Therefore more and more manufacturers consider the introduction of RFID in their production lines. However, RFID projects face various challenges when designing the software infrastructure. In this paper we present design principles which guide application designers in developing and adapting RFID solutions for manufacturing environments. We have derived these design guidelines from a Product Line Analysis (PLA) based on several case studies that we have conducted at manufacturers. Here we systematically identify core assets of RFID applications in manufacturing. Based on the PLA we present a toolset of standard components and their variations along with guidelines for the application design. These guidelines cover processing paradigms for RFID data and heuristics for distributing data and logic in the IT system.

Keywords: Manufacturing, Architecture, RFID.

1 INTRODUCTION

The potential of RFID technology has been widely acknowledged over the past few years. Especially applications in supply chain management are in the focus of attention. Beyond these applications RFID holds potentials for improving manufacturing processes (MCBeath 2006, Chappell et al. 2003). Here, the technology can for instance help avoiding downtimes of production lines or enhancing the acquisition of production data (Günther et al. 2008). Though, for all applications a sound integration of RFID data into the IT environment is a perquisite.

When applying this technology in production, the used IT systems have to be adjusted. Each manufacturer aiming to apply RFID has to deal with the same problem: There exists no guideline how to integrate RFID into the IT infrastructure. The consequence is that the IT department has to develop a solution from scratch, without the foundation of a design framework. This generally increases the complexity of the RFID introduction and hampers RFID investments in the manufacturing domain. Recurring questions include: What functionality is required? What technologies are best suited for implementation? And what is the most efficient distribution of data and logic within the IT infrastructure? Despite each solution is unique, manufacturers deal with some common design structures. In this paper we identify such commonalities and use them to derive general design guidelines. Therefore we have conducted six case studies regarding RFID applications in manufacturing plants. Following the product line engineering approach we derive commonalities and variations for IT systems that support RFID applications in manufacturing. This covers reusable assets including requirements and functional components. Beyond this, we provide guidance for implementing the required functionality and heuristics for mapping the software components to the hardware infrastructure.

The remainder of this paper is structured as follows: Section 2 reviews related work. In section 3 we analyze the manufacturing domain. This includes a short description of the conducted case studies, and today’s IT infrastructure at manufacturers. In section 4 we present results of the completed product line analysis of the manufacturing domain. Here we derive common activities and required data when using RFID. In section 5 we discuss implementation issues, and in section 6 we conclude the paper.

2 RELATED WORK

We categorize existing work on RFID application design into work for horizontal and work for vertical integration of RFID data. The research community put a lot of effort in defining services for horizontally exchanging RFID data. The most prominent set of standards and propositions for enabling services for horizontal integration is the EPCglobal network. Currently specified services in this network comprise interfaces for inter organizational data exchange (EPCglobal 2007) and a lookup service for object related data (EPCglobal 2005). However applications and design guidelines on top of these services target mainly scenarios outside the shop floors of manufacturers (e.g.: Angeles 2005, Främling et al. 2006).

Work on vertical integration of RFID is embedded into the context of RFID middleware (e.g. Bornhövd et al. 2004, Floerkemeier and Lampe 2005, Auto-ID Center 2003). These approaches focus on functionalities and requirements for reader management and data filtering. Here Moon and Yeom (2007) identified abstract functional components that are common for general RFID applications. They identify general interfaces and functionalities. In contrast to all these works we explicitly focus on shop floor applications in manufacturing. For the components that we derive in our domain specific requirement analysis we provide concrete implementation approaches. Unlike existing work our design guidelines also provide heuristics for physically distributing processing logic and required data within the IT infrastructure of manufacturers. These guidelines will ease the integration of RFID systems and middleware components in manufacturing environments.

3 ANALYSIS OF THE MANUFACTURING DOMAIN

In this section we describe the conducted case studies and shortly portray today's IT infrastructure in manufacturing.

3.1 Case Studies for RFID Applications

In order to identify common RFID applications in production plants we have conducted six case studies at large and medium-sized manufacturing companies. (The company names are not revealed due to a nondisclosure agreement.) At all these companies we have evaluated the introduction or extension of RFID applications in production processes. In this subsection we sketch the main applications that we found in each study. The following RFID scenarios are either an improved replacement of barcode technology or scenarios that are not possible with barcode technology at all (e.g. due to environmental conditions on the plant floor). We have investigated the production of airbags, sliding clutches, engine cooling modules, electronic connectors, cast parts, and aluminum foils for packaging.

Airbags: The manufacturer's main driver for considering RFID has been the aim to automate scan processes. Due to legal issues the manufacturer has to register all products at each production step. The backend system then conducts consistency checks on the registered data. Currently, the manufacturer registers the products by reading their barcodes. Yet, due to the shape of the products, workers have to scan the applied barcodes manually at several production steps. These manual scans account for a significant proportion of workers' workload and slow down the production processes. Replacing barcodes with RFID technology could enable to automate all scan processes. This would then increase the productivity of the production lines (Ivantysynova and Ziekow 2007). Another issue is the response time of consistency checks performed in the backend system. The backend system checks for each product whether it passed all preceding production steps and if the assembly is set up with the correct material. In the past the manufacturer has had problems in keeping the response time in the required time frame. However, storing production data on RFID tags at the product would allow the manufacturer to run the consistency checks right at the point of operation and in real time.

Sliding clutches: The manufacturer plans to better narrow recalls with the help of RFID technology. Workers transport materials in the production in stacks of pallets which are manually tracked. The manual tracking is very coarse grained and suffers from inaccuracy. Consequently, the manufacturer does not know exactly when a product is on the shop floor. If a customer detects a production error in the delivered products, the manufacturer needs to call back all possibly affected products. With the current tracking solution this results in large quantities and high costs. Therefore, the manufacturer is interested to improve its tracking system. By tagging the internal transportation units, he could drastically narrow down recalls and reduce related expenses.

Engine cooling modules: This manufacturer applies RFID in a production line due to the demand of a major customer. In the production line they use carriers for the cooling modules which are equipped with RFID tags. The software system of the production line reads routing data from and writes production logs to the tags as the carriers pass different production steps. The production line software communicates with the backend only before the first and after the last production step. By this means, the manufacturer decentralizes the recording of data and reduces the workload of the backend systems. In an extended solution the manufacturer could apply the tags directly on to the cooling devices; thereby the data would always remain with the product.

Electronic connectors: Here the manufacturer has considered RFID for improving the accuracy of tracking production processes. Employees must book each finished production task into the backend system. The backend software is configured in such a way that it only releases booked items for processing in the subsequent step. However, workers tend to forget to book finished tasks. One possible consequence is that subsequent workers have to reconstruct the respective production activity and booked later on. This causes substantial delays in the production process. Alternatively, workers can bypass the IT system. Then the production continues without booking. However this leads to inaccurate data tracks of the production. Here, the manufacturer could automate booking using RFID technology. This would require tagging of all material carriers and installing readers at key positions on the plant floor. In this setup, movements of materials from one process step to another could automatically trigger the corresponding booking transaction.

Cast parts: This manufacturer needs to keep track of several thousand forms that are stored in numerous different stocks. Having the right form available in time is essential for a seamless production. However, the current solution of a manually conducted tracking is inaccurate. Consequently, workers must sometimes search forms before the production can start. Therefore the manufacturer has had to include time buffers for searching in the production process. However, this hampers planning flexibility. The use of RFID could enable automated tracking and thereby increase the tracking accuracy.

Aluminum foils: The dominant reason for the manufacturer to investigate RFID is the decentralization of the IT system. The manufacturer suffered from breakdowns of the backend system in the past. This resulted in production stops of several days. Consequently they aim to decouple parts of the IT's functionalities in the production from functionalities in the backend system. RFID tags on material carriers could enable the desired decentralization. The tags would allow storing routing and production data directly at the carriers. In this setup, breakdowns of the backend system would not impact production tasks that are already released or started.

We conduct our analysis of RFID applications in the manufacturing domain based on the above described case studies.

3.2 Today's IT Infrastructure at Manufacturers

Introducing RFID applications in manufacturing plants requires combining RFID software with existing IT environments in this domain. Therefore we have analyzed the software and hardware systems at manufacturers. In Figure 1 we portray a generalization of architectures that we observed in our case studies. We distinguish between three layers of the IT infrastructure depending on the granularity of the controlled operations. Listed top down, the layers are the backend, edge and device layer. The backend layer comprises the Enterprise Resource Planning (ERP) system and can include parts of the Manufacturing Execution System (MES). Systems in this layer provide coarse grained control and monitoring of the production, e.g. with a temporal granularity of weeks or month. Most functionality of an MES typically resides in the edge layer. This layer realizes more fine grained control of the operations, e.g. the temporal resolution can be days or minutes. The device layer normally comprises the machine software and human machine interfaces (HMI). This layer directly realizes the operations on the plant floor.

The IT hardware of manufacturing is also organized in a layered structure. Here, we can further subdivide the backend layer into local and remote backend. The remote backend comprises hardware that is located outside the plant and accessed via an internet connection. The local backend comprises hardware that is located on the plant floor but within the plants facilities. The edge layer comprises servers and PCs on the plant floor. And the device layer comprises machine controllers (PLC) on the plant floor.

The business information systems and flow of information can generally be described as follows: Sensors on machines generate unfiltered data that are collected by PLCs. A PLC comes with proprietary software. Besides collecting data, the tasks of the software are also to control the machines. The collected data is used for monitoring the actual operations. Additionally PLCs may have links to an HMI. HMIs can display job instructions and may also be used for collecting manual inputs from workers. PLCs pass the collected data on to software on the edge layer. In return, they receive machine configurations from there. The edge layer typically hosts an MES or systems with similar functionalities. The MES is a system which collects data acquired by subsystems, filters this according to predefined business rules and delivers mission-critical information about production activities (Chang et al. 2002, p. 6). It visualizes, optimizes, and coordinates the entire production process in a time frame between days and minutes. In the MES, data are compressed, filtered, and pushed into databases for later analysis, i.e. a historian. Only few data are passed on to the upper level, to the ERP. The ERP conducts the long time business plan and manages the workflow. It passes customer orders down to the MES. From the MES the ERP receives aggregated status reports and information about the material consumption. The ERP does neither directly control machines on the plant floor nor is it involved in production data acquisition. Thus, the functionalities that we address in the remainder of this paper belong to the edge layer and the device layer. For instance the RFID tags and readers would be positioned in the device layer. And the RFID middleware lies in the edge layer. We use our IT infrastructure analysis to point out links with RFID functionalities in section 5.

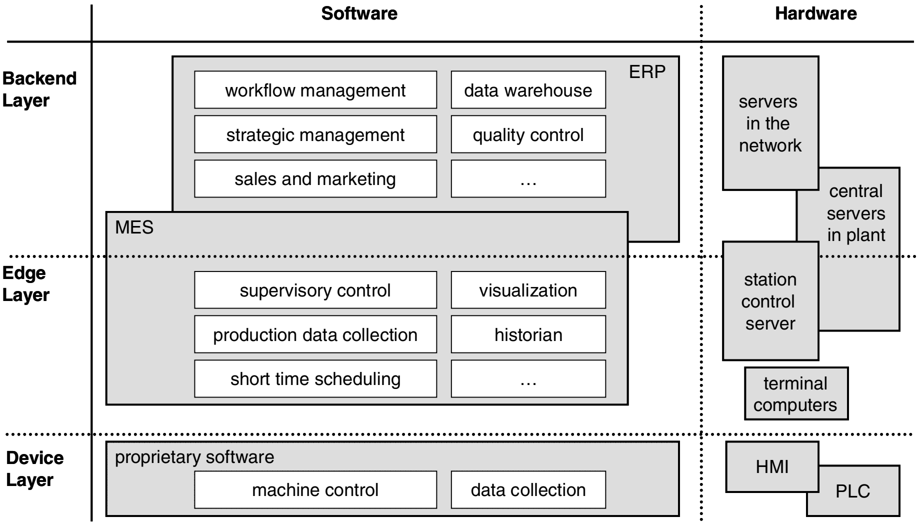

Figure 1. The IT infrastructure in manufacturing.

4 RFID IN MANUFACTURING: COMMON FUNCTIONALITIES

In this section we present our key findings of the domain analysis. We have conducted the analysis based on the method of Moon et al. (2005). Using this method we have identified common activities and variations in IT applications supporting RFID in manufacturing. Our results focus on common activities which a software solution for RFID in manufacturing should generally support. We relate these common activities to the required input data from different software systems. Thereby we provide cornerstones for the architectural design of RFID applications in manufacturing and their integration into existing IT landscapes. The conducted domain analysis comprises four steps: (i) identifying requirements in the targeted domain, (ii) analyzing similarities and variations among the identified requirements, (iii) estimating commonalities of the requirements and (iv) modeling the flow of common activities. Subsequently we discuss each step in detail:

(i) Identifying requirements: In the first step we identify primitive requirements (*PRs*). Moon et al. (2005) define a *PR* as “a transaction that has an effect on an external actor”. For the domain of RFID applications we define external actors as any person or IT system that directly or indirectly uses RFID data. For identifying *RPs* we drew on existing work about RFID in supply chain management (Bornhövd et al. 2004, Moon and Yeom 2007). We then evaluated and extended this initial list of requirements with regards to their applicability to the manufacturing domain. Here we use our case studies from Sect. 2 for identifying the *PRs*: we identify the *PRs* following the flow of RFID data trough the IT system. We list and describe our derived *PRs* in Table 1; we set them in italics.

(ii) Analyzing similarities and variations: In the second step we group *PRs* and if possible generalize them to one common *PR* with variations (in Table 1 non italic requirements). We do this by analyzing the similarity between the *PRs*. We generalize all identified *PRs* based on the semantic similarity of their functionalities. E.g. *low pass filtering* and *statistic filtering* are required activities that can be subsumed into the common requirement: *data filtering* (see Table 2). However, for generalizing *PRs* it is important to consider the level of abstraction on which the respective functionality is applied. Generally, in RFID applications the subsequent processing steps transform the input data to more and more abstract and semantically rich information; having only raw RF data at the beginning and with semantically enriched production information at the end. Therefore, despite semantic similarity of the respective operations, one can not generalize *PRs* for different levels of abstraction. For instance, data cleaning on raw data from an RFID reader is very different from data cleaning of semantically enriched RFID data in a database. Low pass filtering for omitting false positives and negatives is prevalent in the first case. In the second case, one may check the data against logical constraints on attributes. To account for such differences we do not generalize *PRs* for data on different semantic levels.

Table 1: Description of all derived PRs

Requirements	CV Prop. %	Description
RFID reading activity	C / 100%	For read activities we identified three variations. These are manual triggering by plant floor workers (e.g. with mobile readers), regularly scheduled automatic requests, or requests that are triggered by an event (e.g. reads after a light barrier is triggered). This activity may require a reader schedule or a trigger event as input.
manually issuing reads	P / 33%	
scheduled reads	C / 100%	
triggering reads by events	C / 100%	
writing data to tag	P / 67%	The requirement refers to data to the memory of RFID tags.
data enrichment	C / 100%	For enriching RFID data we identified three variations. One is semantic enrichment by e.g. associating the reads with certain process steps. Another requirement is adding reference data such as resolving IDs to corresponding object data. Furthermore streams of dynamic context information can be correlated with RFID read events (e.g. machine sensor data or current machine settings).
semantic enrichment	C / 100%	
stream correlation	C / 100%	
adding reference data	C / 100%	
data filtering	C / 100%	Data filtering refers to removing error from the raw input stream for RFID data. Variations are removal of flickering by low pass filters or complex inference based on statistical filers.
low pass filtering	C / 100%	
statistic filtering	P / 50%	
data cleaning	P / 50%	Similar to data filtering, data cleaning removes errors from input data. The difference is that data cleaning is performed on a higher semantic level and with enriched RFID data. Data cleaning exploits process knowledge to infer on the plausibility of RFID based inputs.
inference	C / 100%	Inference on RFID based events is either done process control or for monitoring purposes. For process control the inference determines a reaction to the RFID input. In monitoring the inferences filters out certain events of interest.
process control	C / 83%	
process monitoring	C / 83%	
notification generation	C / 100%	Activities of notification generation collect assemble, and the information which must be communicated along with detected events. That is, event of interest may require additional information for creating meaningful messages.
aggregation	C / 100%	
adding context information	C / 83%	
message delivery	C / 100%	Notification delivery refers to transmitting captured RFID based events to the desired destinations. This can involve various systems inside or even outside the manufacturing plant. Examples range from triggering processes on the plant floor over providing workers with RFID based information to status updates in the ERP systems.
triggering process step	P / 50%	
reporting asset position	P / 67%	
submitting notifications	P / 50%	
reporting history/status information	C / 100%	

Table 2: Example of a Generalized PR derived from other PRs

Requirements	CV Prop. %	Electronics	Airbags	Packaging	S. Clutches	Cooling F.	Casting
data filtering	C / 100%	√	√	√	√	√	√
low pass filtering	C / 100%	√	√	√	√	√	√
statistic filtering	*P / 50%*	√	√	*X*	*X*	*X*	√

(iii) Estimating commonalities of requirements: In the third step we identify commonalities of the *PRs* based on the frequency of occurrence in the case studies. In Table 2 we exemplarily depict how we derive the commonality of the two *PRs low pass filtering* and *statistic filtering*. The commonality is denoted by the CV property ratio (CV Prob. %). This is the ratio of cases where the requirement exists to the total number of cases. For determining the CV Prop. % we check for each *PR* in how many case studies it occurs. We consider a requirement as common if it occurs in over 80% of the cases (denoted by C in the Table). If it occurs less then 80%, we denote it as an optional property (denoted by P). In Table 2 we depict CV properties for all *PRs*. As it is true for any case-based analysis, we cannot claim that our insights are representative. Though, the significant overlap that we have found in the investigated cases gives confidence that reasonable generalizations can be made based on our sample. Table 2 includes the CV properties that we found for the identified requirements. This provides indication about the commonality of RFID requirements in the manufacturing domain. As the table shows, almost all generalized requirements occur in every investigated application scenario. Thus, it is likely that each RFID based solution in manufacturing will implement at least one variation of functionalities corresponding to these requirements. However, on the level of *PRs* we found variations in the commonality. This means that certain functionalities are not mandatory in the manufacturing domain. The least frequent *PR* that we found (manually reading RFID) occurred in only 2 of the investigated cases. Though most applications do not require manual reading, it is likely that this requirement will occur repetitively in manufacturing applications. Thus, we render this requirement as optional for the manufacturing domain. 11 out of 17 primitive requirements occur in at least 5 of the 6 investigated cases. Given the high frequency of occurrence, we consider these requirements as common for the manufacturing domain.

(iv) Modeling the flow of common activities: In the fourth step we model the common flow of activities and data. Each generalized requirement is denoted as an activity. For each activity we identify the data that is required for performing it. That includes required configuration information and data which is needed in addition to the captured RFID data. By identifying these data items we provide a basis to specify interfaces for integrating RFID applications in existing system landscapes. In figure 2 we present the model of the common flow of activities and data. We also denote required additional input data and the place of its origin (denoted in brackets). Here we point out data sources that can be found in common software systems of manufacturers. This enables situating the common activities in the existing IT landscape.

Throughout the above presented analysis we have identified commonalities among different RFID applications in manufacturing. We analyzed the commonalities (*PRs*) by applying the PLA. Within the manufacturing domain the described *PRs* can be used for deriving IT architectures for each specific RFID application.

5 IMPLEMENTATION ISSUES

In this section we present guidelines for developing RFID software in manufacturing. That is, we provide guidelines for implementing the common activities which we identified in section 4. This includes a discussion about processing paradigms for each of the common activities in section 5.1. In section 5.2 we present heuristics for deciding on the distribution of data and logic within the IT infrastructure.

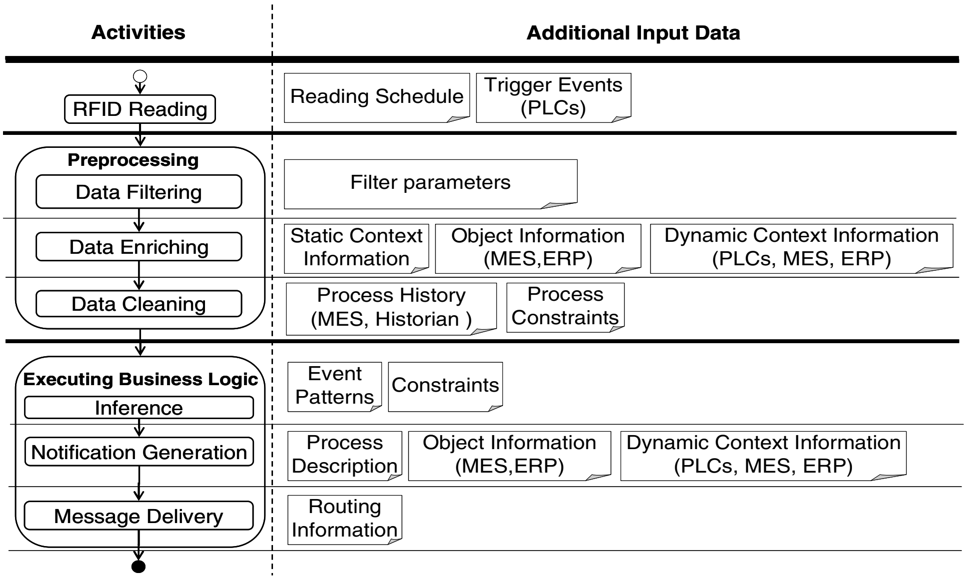

Figure 2. Activities and required data in the capturing process.

5.1 Technologies for Implementing the Common Activities

For processing RFID data we found three classes of common activities. These classes are Reading Activities, Preprocessing, and Executing Business Logic. Note, that manufacturers which do not apply RFID also perform activities in class three: Executing Business Logic (e.g. using manually recorded data). Thus, synergies with existing software solutions (e.g. an MES) may exist. Therefore these existing systems are candidates for performing the RFID specific class three activities as well. Reading and Preprocessing activities are to a large extend RFID specific. These activities require dedicated or specially adapted solutions. Subsequently we discuss technologies for implementing each activity in all three classes in detail.

Reading Activities: The first activity is capturing raw read events, see figure 2. This activity occurs in the device layer. In general we have three options for capturing read events. These are regularly polling read requests, event triggered reads, and manual reads. For regular polling one must create schedules that avoid collisions between readers with overlapping signal fields. E.g. a central control system can avoid signal collisions (Auto-ID Center 2003). Alternatively it is possible to use a decentralized solution where agents on the readers negotiate a schedule (Spiess 2005). For event triggered reads it is necessary to couple readers with an external actuator. Especially in the production environment, activities in the production process can trigger reads: a tag on a material carrier is read out when a machine loads the carrier. However, this requires coupling the machine sensors to the controller of the RFID reader. For manual reads, workers need an interface to readers. This is commonly the case for mobile readers which typically come to use in scenarios with manually issued RFID reads.

Data Filtering Activities: After capturing the raw read events, the data must be filtered. Existing solutions in the device layer use low pass filters to avoid flickering and cancel out double reads (Bornhövd et al. 2004). However, simple low pass filtering with fixed filter windows are not suitable in every application. In our case studies we frequently found conditions on the plant floor that are challenging for RFID installations. This includes presence of metal and spatial proximity of different reading points. This causes the need for more advanced filters that can be configured to account for particularities of plant floor conditions. Such filters must be implemented in the software positioned in the edge layer. Jeffery et al. (2006) have proposed an adaptive filtering solution based on characteristics of the underlying data stream. Brusey et al. (2003) use statistical filters that weight read events to account for particular physical conditions in a specific application scenario. The applied

RFID system should therefore provide the option of embedding a range of highly configurable filtering solutions.

Data Enriching Activities: A simple solution to enriching filtered RFID events with additional data is using a relational database. Here one could dump the incoming data in a table and run batch processes for creating the correlations. This is mainly conducted in the edge layer. However, this approach does not account for the streaming nature of RFID events. One disadvantage of the batch driven paradigm is that it decouples execution time from the arrival time of events. Batch processing is therefore contrary to real time requirements. Furthermore, query languages for databases have poor support for operations on data streams (Babu and Widom 2001). Dedicated stream engines exist that are designed for real time correlation of data streams with additional data (Abadi et al. 2003). These approaches match the requirements for the data enriching activity. Stream operators can correlate RFID read events with static information about their sources and dynamic information from machine sensors. Expressive high level languages for defining such stream operations are available (e.g. Arasu et al. 2006). Thus, the stream oriented processing paradigm is suitable for realizing the activity of data enrichment; especially for applications with real time requirements.

Data Cleaning Activities: This activity is for checking the plausibility of detected events and making corrections where possible. For instance, previous scans of the pallet can allow completing missing reads in pallet scans. Also, duplicate reads from different sensors may be dropped. Jeffer et al. present a framework for such cleaning based on stream processing operations and predicate checks (Jeffery et al. 2006). Here, the predicates implicitly encode process knowledge. The peculiar conditions on many plant floors make cleaning steps on this level necessary. Software systems for RFID applications in this domain should therefore support the definition on semantic rich rules for constraint based data cleaning. This software would normally lie in the edge layer.

Pattern Matching and Notification Generation Activities: The fifth activity in the flow of common activities is inference on input data for detecting events of interest. The sixth is for generating corresponding notifications. From the computational point of view these are different processing tasks that are conducted in the edge layer. Yet, techniques of complex event processing (CEP) (Luckham 2001) cover both tasks. CEP is of relevance whenever the events of interest results from set of input events rather than from single events. The semantic aggregation of CEP is necessary to avoid information overload. This applies specifically to notifications to backend layer, e.g. notifications from the MES to the ERP. Thus, direct support for defining and running CEP rules within the RFID solution is beneficial in many manufacturing applications. CEP evaluates rules which define an event pattern, constraints on the pattern, and an action. In that sense CEP is similar to ECA rules in active data bases. However, CEP rules can be triggered by arbitrary event messages and are not limited to database specific events (such as insert or update). This supports formulation event patterns on RFID events captured in a production process and fits to the event driven nature of RFID data. Using state machines for complex event processing (Gehani et all. 1992) or adapted Petri nets (Gatziu and Dittrich 1993) are options for implementing this kind of pattern evaluation.

Notification Delivery Activities: The last activity for RFID data capturing is delivering notification messages to the respective addressees in all three layers. Here we can distinguish four different communication styles. The dimensions for the distinction are push vs. pull based communication and direct v.s indirect addressing schemes. Periodic pull based loading of data chunks in the manner of an ETL process is appropriate for analytic applications. Though, push based communication is a better fit to the event driven generation of notification messages. Furthermore, monitoring and control applications need more timely updates. This applies in particular if control applications directly steer operations on the plant floor. These are arguments for applying the push based communication scheme for RFID data acquisition at manufacturers. Both, push based and pull based communication can be combined with direct or indirect addressing. Direct addressing requires to explicitly denote the destinations of each notification type. Indirect addressing via an intermediate service decouples data provision from data consumption. This decoupling enables flexible adaptation of notification delivery because data sources and recipients can be managed separately. Event based systems use a notification service and publish/subscribe mechanisms combine the push based communication scheme with indirect addressing (Mühl et al. 2006). Using such a notification service matches to the requirements of many of the investigated application scenarios and should therefore be available in RFID enabled IT infrastructures for manufacturing.

5.2 Distributing Data and Logic

In section 4 we discussed common functionalities and their implementation in section 5.1. Now we provide heuristics for decisions on the distribution of the functional components. This goes in hand with the distribution of corresponding input and output information. Writable memory of RFID tags enables pushing data management down to lower system layers. RFID memory can store production data – such as recipes or production records – right at an object (e.g. the product). This allows running several operations solely in the edge layer, i.e. consistency checks can run locally on station control servers or terminal computers without backend interactions. However, throughout our case studies we found reoccurring arguments for as well as against decentralization. Table 3 and 4 list the identified key benefits of decentralization and drawbacks respectively.

Table 3: Benefits of Decentralization

Scalability	Decentralized systems distribute the total workload on several devices is the system. As the system grows, the number of devices that share a task grows as well. Furthermore, pushing aggregation and filter functions down the hierarchy to the information sources reduces network traffic and avoids overloading higher system layers.
Performance	Performance bottlenecks in the IT can impact the productivity on the plant floor. We found that the tolerable delay is typically about 0.5 sec for manually issued request and in the order of milliseconds for machines. The cause for longer delays can be delays for accessing remote systems or peaks in the workload. Placing the logic and required data close to the point of operation avoids the problem of network delay. Furthermore, decentralization of processing tasks mitigates peaks in the workload.
Reliability	System failures that cause a breakdown of the production can account for major expenses. Decentralization can help to avoid single points of failure and limit the number of affected production tasks.
Eased data association with objects	Through the RFID enabled decentralized information storage it is ensured that the information is available at the point of operation. This helps to avoid that wrong information is associated with an object.

Table 4: Drawbacks of Decentralization

Redundancy	A lot data that is collected on plant floors is not only used for steering the production but also to document the conducted steps. That is, a certain percentage of the data must be available in the backend system. Distributing this data to lower system layers consequently results in a certain degree of redundancy.
Maintainability	Distributing logic across system layers adds complexity to the management of the IT systems. This is because maintenance tasks cover systems that run at several locations and possibly different platforms. If most functionality resides on a central system, the majority of maintenance tasks are done on the same platform.
Inconsistency	Information on tags can only be read if the item is in the proximity of a reader. Thus, for applications that require data access independently from the corresponding object, it is necessary to have the information available in the network as well. Keeping data redundantly on the tags and in the network bears the risk of inconsistency.

Application designers can decide on the distribution of data and logic within the system along the in table 3 and 4 listed arguments. It depends on the particular application how much these arguments weigh. In our case studies we found that RFID applications fall in two major categories regarding the trade-off between centralization and decentralization. These categories are applications for *monitoring/analyzing* and *controlling/steering* processes. For each application category we derive guidelines for distributing processing functionally. We base our analysis on the three previously identified main activities: RFID Reading, Preprocessing and Executing Business Logic (see Figure 2)

Applications for *controlling/steering* processes perform checks in the running processes and steer the operations on the plant floor. These applications commonly have strong real time constraints. That is, the production processes slow down if machines or workers must wait for responses from the IT system (e.g. Ivantysynova and Ziekow 2007). Thus, performance is typically the main concern. Consequently, it is often desirable to run the three activities RFID Reading, Preprocessing and Executing Business Logic in the edge layer and without backend interaction. The guideline is to push the operations as close to the physical processes as possible.

Applications for *monitoring/analyzing* processes support management decisions and help spotting potentials for improvements in the production processes. Here, no real time requirements apply. Typically, solutions like data warehouses or reporting tools extract, process, and visualize data from several sources on the plant floor. Such applications are often integrated in MES or ERP systems. Consequently, the activities of Executing Business Logic for monitoring/analyzing processes with RFID should run in the backend (see Figure 1). However, in order to ensure scalability it is desirable to reduce the input data on lower system layers; i.e. the edge layer. Particularly data cleaning and data filtering (part of the activity Preprocessing) reduce the amount of transmitted RFID data. Filtering raw read events can take place within device controllers (Bornhövd et al. 2004). These may run on terminal PCs in the edge layer. Data enriching and data cleaning (part of the activity Preprocessing) can require information from the backend and other sensors on the plant floor (see Figure 2). Such context information is particularly relevant for monitoring and analysing production processes. A suitable placement of enriching and cleaning operations is therefore where RFID data and required context information join. This point depends on the targeted system environment. Yet, we found that station control servers in the edge layer are suitable for enriching and cleaning operations in many cases. Therefore the guideline is to join information as close to their sources as possible and run operators of data enriching and data cleaning at this point.

Note that design decisions on the distribution of data and logic are heavily dependent on the targeted manufacturing environment. In our case studies we found that hardware infrastructures at manufacturers are very heterogeneous. That is, not every device type depicted in Figure 1 is always in place; e.g. not every manufacturer has terminal PCs on the plant floor or station control servers for controlling several machines. Yet, most IT systems at manufacturers have some sort of a hierarchically structured hardware landscape. Our guidelines help placing operations for RFID data processing into this hierarchy.

6 CONCLUSION

Our aim was to analyze common aspects of RFID applications in the manufacturing domain and to use them for design guidelines. These guidelines are tailored to RFID applications and respect the particularities of manufacturing. This includes typical applications as well as typical IT systems in this domain. In order to develop our design guidelines for this application domain we have conducted six case studies regarding RFID applications in manufacturing plants. Focusing on production specific issues we have identified core components and technology paradigms for RFID applications in manufacturing. Following a product line engineering approach we derive common activities and variation points for respective RFID infrastructures. We did this with explicitly respecting the IT-environments as they are typically found in manufacturing plants. That is, we relate our components to existing software systems, showing where RFID data processing can fit in.

Beyond defining functional components, we have investigated which processing paradigms and technologies are most suitable for challenges in manufacturing environments. Furthermore, we discuss the mapping of software components to hardware in the infrastructure. This is because RFID enables novel distributions of data and logic, which is a driver for using RFID in several cases. Overall the design guidelines presented in this paper comprise aspects of (i) functional components, (ii) interfaces and interaction with external components, (iii) processing paradigms, and (iv) distribution of data and logic.

References

Abadi, D., Carney, D. , Cetintemel, U., Cherniack, M., Convey, C., Lee, S., Stonebraker, M., Tatbul, N., and Zdonik, S. (2003). Aurora: a new model and architecture for data stream management, The VLDB Journal, 12, (2), 120–139.

Angeles, R. (2005). RFID Technologies: Supply-Chain Applications and Implementation Issues. Information Systems Management, 22 (1), 51-65.

Bornhövd, C., Lin, T., Haller, S., and Schaper, J. (2004). Integrating automatic data acquisition with business processes - experiences with sap's auto-id infrastructure. In: Very Large Data Bases Conference, 1182–1188.

Brusey, J., Floerkemeier, C., Harrison, M., and Fletcher, M., (2003). Reasoning About Uncertainty in Location Identification with RFID, Workshop on Reasoning with Uncertainty in Robotics at IJCAI.

Chang, Y., McFarlane, D., Koh, R., Floerkmeier, C. and Putta, L. (2002). Methodologies for Integrating Auto-ID Data with existing Manufacturing Business Information Systems. White Paper, Auto-ID Labs.

Chappell, G., Ginsburg, L., Schmidt, P., Smith, J., and Tobolski, J. (2003). Auto-ID on the Line: The Value of Auto-ID Technology in Manufacturing. White paper, Auto-ID Center Massachusetts Institute of Technology, Cambridge.

EPCglobal (2005). Object Naming Service (ONS) Version 1.0. White paper. http://www.epcglobalinc.org/standards/ons (last accessed February-18-2008)

EPCglobal (2007). Epc information services (epcis) version 1.0 specification. White paper.
http://www.epcglobalinc.org/standards/epcis/ (last accessed February-18-2008).

Floerkemeier, C. and Lampe, M. (2005). RFID middleware design: addressing application requirements and RFID constraints. sOc- EUSAI '05, 121, 219-224.

Främling, K., Harrison, M., and Brusey, J. (2006). Globally unique product identifiers - requirements and solutions to product lifecycle management. 12th IFAC Symposium on Information Control Problems in Manufacturing, 855-860.

Gatziu, S and Dittrich, K.R. (1993). Events in an active object-oriented database system. International Conference on Rules in Database Systems, 23-39.

Gehani, N.H., Jagadish, H.V., and Shmueli, O. (1992).Composite event specification in active databases: Model and implementation. Very Large Data Bases Conference, 327-338.

Goyal, A. (2003). Savant Guide. Technical Report, Auto-ID Center Massachusetts Institute of Technology, Cambridge.

Günther, O., Kletti, W., Kubach, U. (2008). RFID in Manufacturing. Springer. Berlin.

Ivantysynova, L., Ziekow, H. (2007). RFID in der Produktion: Eine Fallstudie aus der Airbagindustrie. GI Workshop RFID-Einsatz in kleinen und mittelständischen Unternehmen at the GI conference.

Jeffery, S., Alonso, G., Franklin, M., Hong, W., and Widom, J. (2006). Declarative Support for Sensor Data Cleaning. Pervasive, 83-100.

Luckham, D. C. (2001). The Power of Events: an Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley Longman Publishing Co., Inc.

MCBeath, B. (2006). RFID for Manufacturers: How Manufacturers are Improving Processes by Using RFID. White paper, ChainLink Research.

Moon, M., Yeom, K., and Chae, H. S. (2005). An Approach to Developing Domain Requirements as a Core Asset Based on Commonality and Variability Analysis in a Product Line. IEEE Transactions on Software Engineering, 31 (7), 551-569.

Moon, M. Yeom, K. (2007). Product Line Architecture for RFID-Enabled Applications. BIS, 638-651.

Mühl, G., Fiege, L., and Pietzuch, P. (2006). Distributed Event-Based Systems, Springer.

Shivnath, B. and Widom, J. (2001). Continuous Queries over Data Streams, SIGMOD Record. Sept.

Spiess, P. (2005). Collaborative Business Items: Decomposing Business Process Services for Execution of Business Logic on the Item, European Workshop on Wireless Sensor Networks.