Embedded RFID Middleware
The interface between RFID systems and enterprise applications

Embedded Innovation GmbH & Co. KG, Steinkaule 86, 53757 Sankt Augustin
Michael.Wack@embeddedinnovation.de, Bernd.Schmidt@embeddedinnovation.de

1. Abstract

RFID (Radio Frequency Identification) is the latest technology for automatic identification systems that already play a major role in industry, particularly in logistics applications. This paper describes the Software Architecture of Embedded RFID Middleware (ERM) and the advantages of using such an embedded system. ERM facilitates the integration of RFID systems in existing enterprise applications.

2. Introduction

RFID (Radio Frequency Identification) is the generic term for the contactless identification of objects on the basis of radio frequencies. For this reason RFID is the synonym for system solutions to global and local tracking and tracing by the use of transponders. Together with bar code and other technologies RFID belongs to the Auto-ID procedures, i.e. ways for identification of objects without human interaction. It closes the gaps towards IT systems, which had to be bridged by manual data input.

The use of RFID systems in supply chain management and industrial automation has recently stepped up. Especially in these domains the RFID technology keeps the promise to eliminate many existing business problems by bridging the economically costly gap between the virtual world (IT systems) and the real world of products and logistical units. Common benefits include more efficient material handling processes, elimination of manual inventory taking, and the automatic detection of empty shelves and expired products in retail stores [1, 2, 3]. However, it is not just the business community that can benefit from the use of RFID technology, but also the consumer.

The widespread adoption of RFID requires not only transponders and readers, but also the appropriate networking infrastructure. The implementation of RFID makes additional demands on the networking structure. The increasing number of terminals and the additional rise of data are counted among this. Apart from that, the implementation of RFID means the direct integration in business processes. This results in new demands in respect of the availability and the architecture for communication.

Such a supporting RFID infrastructure typically comprises a component often referred to as RFID middleware or edgeware. RFID middleware, simply put, is a software layer residing between the RFID hardware and the existing back-end system or application software. It extracts data from the RFID readers, filters it, aggregates it and routes it to enterprise applications such as a warehouse management system (WMS), enterprise resource planning (ERP) software or a manufacturing execution system (MES).

The main contribution of this paper is an embedded middleware design that addresses both, application needs and the constraints of RFID technology. To do that, we analyse the requirements the RFID software component should meet, in order to manage large deployments of readers and the amount of data these readers capture.

The paper is organised as follows: Section 3 provides a brief overview of RFID technology and outlines the constraints imposed by the characteristics of RFID as well as the application requirements RFID middleware should meet. In Section 4, we discuss the Software Architecture of Embedded RFID Middleware (ERM), an RFID middleware platform that addresses the requirements and constraints outlined in the previous sections. The paper concludes with a summary of our contribution in Section 5.

3. RFID Technology

RFID uses radio waves to establish communication links between RFID reader devices and transponders (also called "labels" or "tags"), which could be attached to any objects or persons. The tags can be passive, semi-passive or active and there are read-only and read-write tags. The storage capacity in case of read-only labels is usually 8 - 100 Byte, in case of rewritable labels up to several Kbytes. Some of the “magic” of RFID is that the data associated with an object can be captured automatically without human intervention or line-of-sight between the RFID reader and the object’s tag. The ability of an RFID reader to record observations of hundreds of uniquely tagged objects per second creates myriad opportunities for businesses [2].

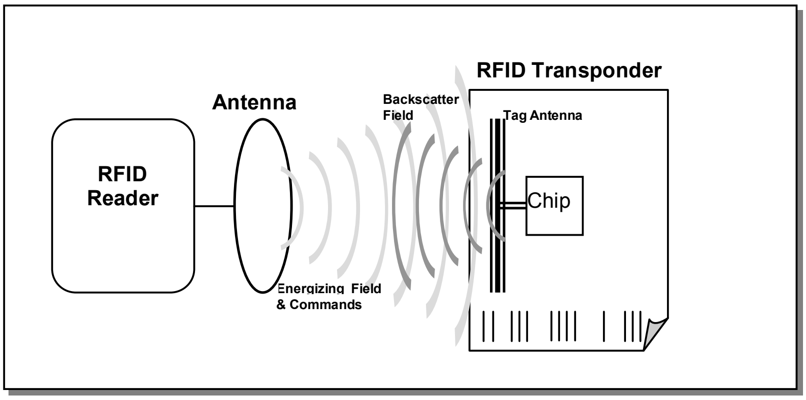

Figure 1: Typical RFID components

The typical components of an RFID system are shown in figure 1. An RFID transponder is a microchip with memory and a connected antenna, usually embedded into some kind of label and applied to an object. The microchip typically contains functional blocks such as a switch, a demodulation module, memory, control logic and a voltage converter. Passive tags do not have a built-in power source or battery. Instead, the RFID reader is required to transmit a carrier wave that effectively powers the tag. Modulating some of these carrier wave transmissions are commands, which after demodulation can in turn activate the tag’s control logic, which might short out the antenna in a predetermined sequence to communicate to the reader an identifier, contained in tag memory. This communication takes place via a technique commonly referred to as backscatter modulation.

The main differences in existing RFID systems are the frequency band, coding and modulation. These factors result in specific characteristics like reading range, reading rate, transmission speed, sensitivity to material and environment and of course cost.

In summary, the advantages of RFID among others are contactless identification at distances up to several hundred meters without the need to have intervisibility, reading and writing of data, high reading rates and reading of closely located objects, e.g. a pile of boxes, independent of the orientation of the label (bulk reading).

Industrial areas often come along with harsh environments (extreme temperature ranges, high humidity, vibrations, shocks, chemicals), limited space as well as real time constraints. Naturally, these environments are unsuitable for standard PCs. Such conditions call for robust, compact systems with high availability, like embedded devices. Furthermore, the huge raw data rate produced by RFID readers requires real-time processing which typically only an embedded device with a real-time operating system can cope with.

In the UHF range and the desired high reading ranges the circumstances are considerably more complex. Objects behind the tag or worse, in the measuring field, affect the whole reader-to-tag system. The effects vary from simple attenuation beyond reflection and refraction up to scattering of the electromagnetic wave. Due to the variety of possible effects, there should be no objects between reader and tag during the reading operation. Exceptions may be thin, non-conductive layers as well as materials with low attenuation and low dielectric coefficient like dry paper, loose matters or expanded polystyrene. But also outside influences must be considered. Tags may happen to be read if they are situated below a dry canvas cover. However, if the same canvas cover is wet, the probability is high that those tags cannot be read.

Some of the RFID-enabled benefits being pursued include improved retail-product-availability, reduced inventory levels, streamlined operations, automated audit trails, and a broad range of track-and-trace related applications. A well-known example for using RFID technology was the access control for the 2006 Soccer World Championship in Germany. Other examples are hospitals which are deploying RFID to track patients and equipment and aerospace companies tagging parts and assemblies to track them both during manufacturing and throughout their operating lifecycles [3, 4].

3.1 Impacts on Enterprise Applications

The impact of RFID technology on enterprise applications is more complex than the simple connection of a new class of network device. The following issues have to be solved by RFID middleware in order to come to a successful RFID deployment.

Incompatible Readers
Readers from different manufacturers require their own instruction set and their distinct protocol to be controlled and to deliver raw data. This prevents customers to easily deploy RFID components from different suppliers in the same system. Furthermore, readers have different networking capabilities, starting with a simple RS232 interface through to a full-featured TCP/IP host interface.

Lack of Sensors and Actuators
Readers on the market usually only provide very basic digital inputs and outputs that can be switched to control a light or acoustic signal. Depending on the business context, more sophisticated sensors and actuators may be required which could be provided by an embedded system, like a PLC.

Different Enterprise applications
Often the information from the RFID system is of interest not only to a single application, but to a set of different applications across an organisation. The variety of different business applications calls for various RFID application interfaces (API) depending on the business context, e.g. distinct APIs for automation, automotive, transport and logistics. Moreover, different latencies shall be supported, depending on the application type. Some applications need an immediate interaction with the object, which requires a real-time latency. On the other hand, e.g. a legacy application might only receive batched updates on a regular basis.

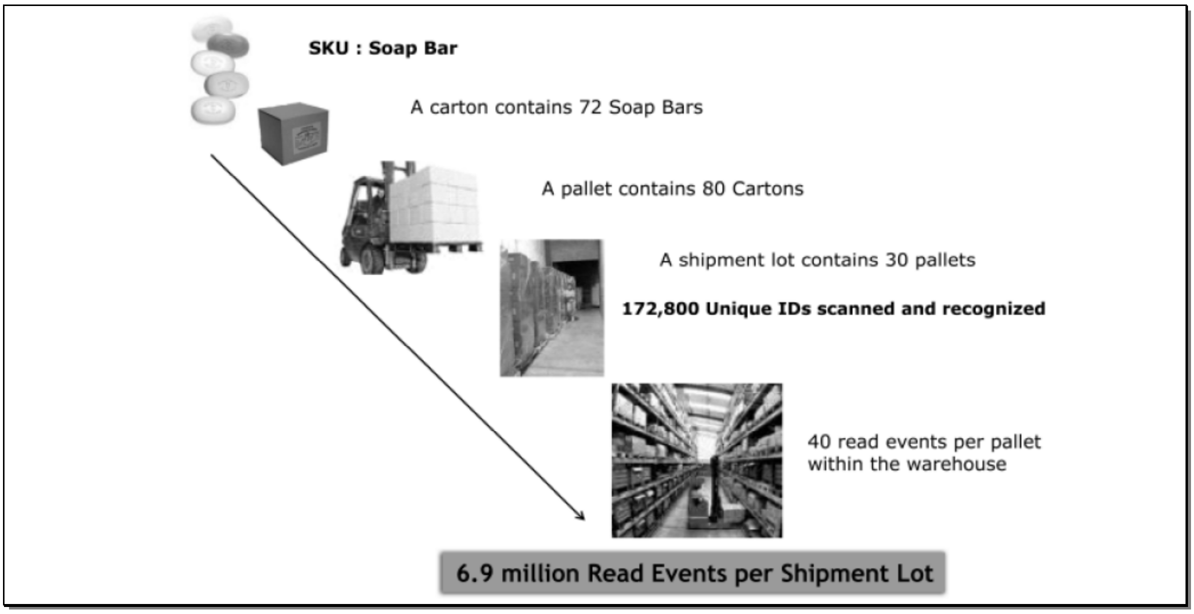

Figure 2: RFID Data Explosion

Flood of raw data

RFID readers continuously detect and read tags that are in range and therefore produce a huge amount of raw data, multiple reads of the same tag and hence redundant data included. As illustrated in figure 2 processing these data would put an unbearable load on enterprise applications [6]. Obviously, the raw data must be filtered and transformed into data, which are meaningful to the enterprise application. Moreover, since RFID allows the identification on the instance level and not only on the class level, like e.g. bar-code, data and information needs to be aggregated accordingly, if an application can not or do not want to handle the increased granularity.

IT-service management

The integration of RFID readers into an existing IT infrastructure needs special support for change and configuration management, which is usually not covered through traditional IT Systems.

Real-time management

Things happen in real life processes. It is a supplier warehouse, a forklift, a customer dock door or a conveyer belt. It is important to know that when we discuss the ability to apply RFID technologies, we virtually always imply management of data and material in real-time. This necessitates wireless technologies that integrate into a coherent architecture of computing. The concept of “edge” computing in turn necessitates a whole new way of thinking; it will also require a new approach to management in general. In general, management is about making the right decisions. Today, it not only continues to be important that we make decisions at the right time, but much more so that we make them in the right time. Over the coming years, we will witness an evolution towards faster decision-making and, along with it, decision-making that will shift towards the edge of the enterprise, simply because these new technologies allow for faster, cheaper and better business processes. They lead to competitive advantages that accelerate individual companies in industry after industry.

3.2 Impacts by RFID technology

Beside the impacts described above, there are impacts from the technology itself, due to the low-power and low-cost constraints of RFID transponders.

Read probability
There is no 100% *probability* that a specific transponder is detected and identified at a particular time, even if the object with the tag attached is in the read range of a reader. There are different physical reasons for that problem:

> *Environment*: The radio wave could be eliminated through absorption, reflection or refraction. This could happen through objects or the environment in the range of the reader, e.g. metals or liquids.
> *Interferences*: Through multipath scattering there could be interferences, which annihilate the radio wave.
> *Collisions*: If several tags are activated through the reader, it is common that collisions happen on the air interface if tags send their information at the same time. Even anti-collision protocols could not prevent read failures in all cases.
> *Velocity*: The faster an object with attached transponder is moving, the less is the probability that this tag is detected.

Regulation: Due to (the European) radio regulation there is only a limited communication bandwidth. For example in the UHF band the regulation permits the use of fifteen 200 kHz-wide channels between 865.0 MHz and 868.0 MHz. Additionally, the readers need to implement the "listen before talk" scheme [5]. This means, that readers need to listen for other transmitters using the channel before beginning to communicate. If used not properly, a passing tag could be missed, because the reader is not operating at that time. Furthermore, several readers which are close to each other need to be coordinated to prevent a deadlock situation.

Bandwidth: The bandwidth limits the transmission rate in the air protocol, which in turn limits the total number of tags which can be detected per time slot.

3.3 RFID Standardisation

EPCGlobal, a member-driven organisation, is leading the development of industry-driven standards for the Electronic Product Code (EPC) to support the use of Radio Frequency Identification. There do already exist standards concerning the RFID frequency band and the air interface [5, 7]. However, an approach of standardisation, as came about for the retail market, was not pursued by the EPCglobal consortium for other domains like e.g. production. The VDA recently established working groups for RFID, but not with the same efforts compared to big retailers like e.g. Metro, Walmart or Tesco. For the VDA the standardisation of handling and management of containers has been of main interest so far. Additionally, further standardisation among VDA and EPCgobal are ongoing, which are of interest for the majority of suppliers in the automotive industry. This especially applies to the introduction of closed loop applications for the line of production.

The standardisation process, driven by EPCglobal, is not only focused on data standards but also addresses the air interface and the protocol domain. Apart from that, the method of reading and counting Electronic Product Codes in a reader environment according to various criteria is defined with the Application-Level-Events (ALE) specification, but needs to be refined in order to address the requirements of different markets [7]. The specification includes a formal processing model, an Application Programming Interface (API) described abstractly via UML and bindings of the API to a WS-I-compliant SOAP protocol with associated bindings of the key data types to XML schema. WS-I is an open industry organization chartered to promote Web services interoperability across platforms, operating systems and programming languages [8].

However, the ERM software has to support upcoming standards as well as legacy interfaces to support current products e.g. RFID readers, printers as well as simple embedded applications.

4. Embedded RFID Middleware

The availability of RFID systems depends on all components, whereby the RFID middleware plays a crucial role. This relates to the reliability, the security and availability of the data and the scalability of the systems. This section describes the software architecture for the embedded RFID middleware ERM, where the requirements and constraints can be reliably solved. To come to a robust solution, a structured approach based on a real-time operating system as shown in figure 3 is mandatory.

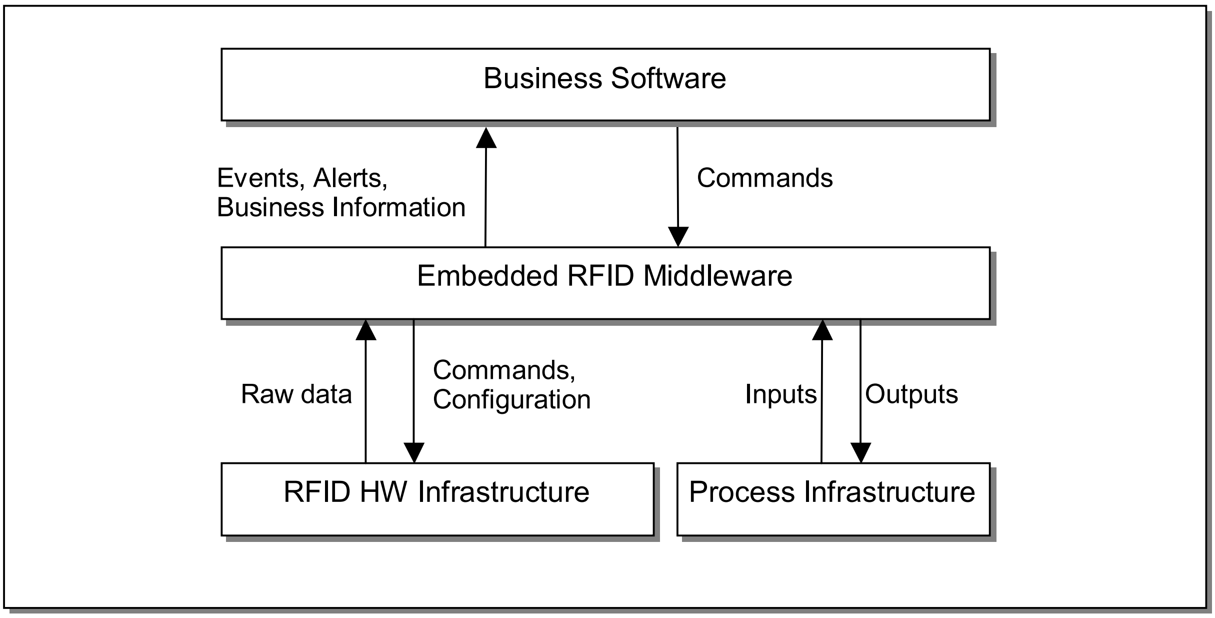

Figure 3: Main RFID software components

Figure 3 gives an overview of the main software components of an RFID system. The RFID hardware infrastructure is mainly composed of transponders, readers, antennas and optional printers. These devices need to be managed, e.g. for setting up configurations, and are controlled by means of dedicated commands. They usually deliver raw and unprocessed data in real-time.
The Process Infrastructure component is optional and consists of sensors and actors. They can be directly controlled by the ERM or through a PLC system which communicates with the ERM layer.

The tasks of the ERM software are described in section 3. The components and the structure are presented in this section.

Through a standardised API the Business Processes are connected to the ERM software. On one hand, there is a set of commands which enable the Business Processes to manage and control the RFID system according to the application requirements. On the other hand, the ERM sends business related information e.g. through events or alerts back to the application. Since to communication will be handled in terms of an API, the ERM supports all types of applications. The Business software could be a back end system such as a warehouse management system (WMS), enterprise resource planning (ERP) system or a manufacturing execution system (MES) or a combination of such systems.

4.1 ERM layer

The ERM is a structured configuration of individual components with dedicated tasks which are interconnected through an internal virtual event bus. Each component uses exactly that information from the event bus which is needed to fulfil the dedicated task and in turn to produce results, which are placed on the bus system. These results can then be part of the input for other components.

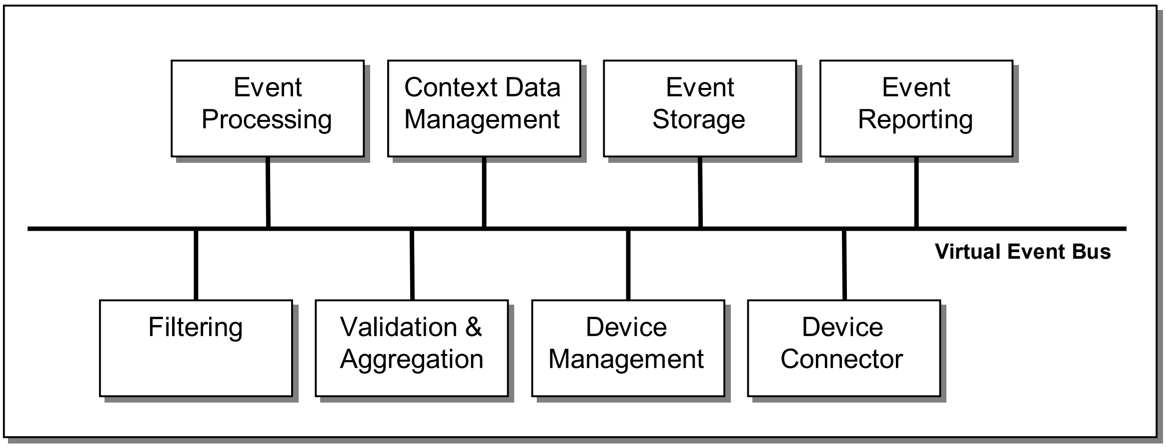

Figure 4: ERM component architecture

Depending on the complexity, each component can be realised by one or more processes. The smallest item which can be processed by the ERM system is an event. Such an event includes the tag ID, date, time, reader ID, the location where the reading took place and an indication of the direction.
There is no predetermined order or schedule for the components. This solely depends on the inputs one component needs. In this way, the computational order of precedence forms a pipeline with an event at the beginning and an eventually complex result at the end. Since there are no direct calls between two components, it is possible to create a distributed system, where some components run on different nodes. This can be necessary, e.g. for the event storage where a greater amount of RAM or disk space is needed.
These architecture results in a high performance system, which is scalable, portable and modular. In order to meet real-time constraints, the underlying operating system shall of course have a real-time characteristic.

4.2 ERM components

In the following section a brief description is given to the different types of components shown in figure 4.

Filtering: This component allows filtering out some read events. Usually this means to remove read events from dedicated readers, remove a certain tag identifier or tags which contain specific data. But also to filter data according to predefined constraints with global and domain information is a task. For example, multiple readers can generate duplicate readings. To filter this, a filter will scan data within a sliding window to find if there are duplicate readings from multiple readers, and delete the duplicate if it exists.

Validation & Aggregation: The purpose of this component is to reduce the overall number of events. Events can be aggregated into semantic data, such as containment relationships. Associating relationship among objects has been identified as a difficult issue for RFID applications. This can be done by several strategies, depending on the application. For example, it is common that an application is only interested in an event when an object (with attached transponder) enters or leaves a certain area. Other applications only want a total count of objects (hence the total count of transponders). Beside that, it is mandatory to validate the reads and the state of the transponder to overcome the problems with the Read probability.

Device Management: In order to integrate readers in the IT infrastructure, there must be means to manage the reader configurations, manage firmware downloads and support the change of defect readers.

Device Connector: This component realises the concrete connection from a reader to the embedded system e.g. by RS232 interface, CAN Bus or TCP/IP. Here, a device health monitor can be established in an effective way. Since the component knows all readers, the coordination, which antenna will be switched on or off to get an optimal performance and lower the interferences, will be realised here. Additionally, the communication with other systems via fieldbuses and the control of sensors and actuators is handled by this component.

Event Processing: Through this component, the ERM layer communicates with one or more applications. This can be realised best by a publisher/subscriber method, where the subscriber defines in which data (events) he is interested. The component has also to consider the different latencies, the applications need. The scope ranges from real-time to batch processing. Together with the other components, this layer enables the business processes and solutions to use the RFID information that was generated in real-time. This component is where other services or business partners, and the RFID system are integrated.

Context Data Management: After the Filtering and the Aggregation had happen, usually an event can be published on the virtual event bus, to inform potential subscribers. RFID events are similar in size of stock trades—they're very small events with no intelligence in the information itself, so there is a need to provide context to the event through process or exceptions. It requires context to become information. That means you will need to 1) augment this data with meaning, 2) support it with relevant location- and time-specific details from internal applications, third-party systems and Web services, and 3) tie this information to specific business processes. Finally, in order to translate this into actionable intelligence, this information must be aggregated, summarized into meaningful reports and distributed to various touchpoints along the information supply chain (ISC).

Event Storage: The ERM must be able to store the events in an Event Database in order to be able to control and measure the events during the whole process. Based on the history of events, business and process decisions can be established.
Event reporting: The reporting component supports applications to get data from the Event Database with a standard interface. For example, one query can find all tags coming from one reader during a time interval or find all readers which read a specific tag.

Virtual Event Bus: All components are connected to and communicate with the virtual event bus service. The system is essentially event-driven as changes of state in resources need to be notified asynchronously to several, potentially unknown, recipients. An event may indicate the discovery of a new transponder, a component failure, and the change in context or condition. We have chosen to implement the event bus as an at-most-once, persistent publish/subscribe delivery service, using a router to distribute events to subscribers in which

the service attempts to deliver the event until it knows that the subscriber is no longer a member of the system.
The router is content-based – i.e., a subscriber specifies a filter when it registers, and all published events that match the filter are forwarded to that subscriber. Publishers do not need to register with the Router. When a publisher sends an event to the Router, it does so synchronously and reliably. Successful delivery of the event to the Router transfers responsibility for subsequent delivery to the Router. The Router attempts to deliver such an event to each subscriber whose filter matches the event. If it is unable to deliver the event to a particular subscriber due to transient communication failure, it queues the event for redelivery to that subscriber. The router attempts to deliver queued events until it knows that the subscriber is no longer a member of the system. Each subscriber is guaranteed to receive all events from a particular publisher in the same order as received by the Router. This is required in case there is a causal relationship between events from a particular publisher. If the Router receives a component-left event, it removes that subscriber's filters (if it had registered for any events), and purges any queued-up events for that subscriber.

4.3 ERM Controller

To manage all requirements and to implement the architecture for the ERM software, an appropriate hardware is needed, too. ERM software will be implemented on a dedicated machine that can be stand-alone or implemented into an existing embedded system, if the available processing power is sufficient. The definition of such an embedded system is as follows: An embedded system is a Software/Hardware unit that is linked to an overall system through sensors and actuators. Within this overall system the embedded system assumes observation, control respectively feedback control tasks. As a general rule, embedded systems are a matter of reactive, frequently hybrid, distributed systems with real-time demands. Typically such systems are not directly visible for the human user. He interacts unconsciously with the embedded system. A reactive system translates input events (which frequently occur stochastically) - a lot of times under compliance with time allowances - into output events.

A so-called ERM controller is the physical implementation of such an embedded system, running an instance of the ERM software. Depending on the application, one or more ERM controllers are needed to manage all requirements. Since the ERM is portable and not hardware-dependent, it can be implemented on all suitable hardware platforms, as long as the specifications e.g. temperature ranges or fan- and head-less operation are fulfilled.

5. Summary

RFID technology is significantly changing the current business applications. Many industries will consider, test, and deploy RFID technology as part of their automated data collection infrastructure. The potential benefits are compelling. As RFID proliferates and production deployments roll out, there will be significant impacts to consider and address from an enterprise networking perspective. One of the major challenges for RFID technology is RFID data management.
In this paper, we have identified several key issues in managing RFID data, including data modelling, automatic data transformation and enrichment, effective support of queries for tracking and monitoring, scalable data archiving, and convenient application integration. ERM provides a standard control interface to integrate RFID systems into existing enterprise applications and application-conform RFID data. The advantages are reduced cost and time for integration as well as for exchanging existing RFID readers or components. On top of that, ERM stands for reliable object identification due to its capability of data processing in real-time as well as for the control of actuators.

6. References

[1] RFID-Handbuch; Finkenzeller, Klaus;
Hanser Verlag; ISBN 3-44622071-2, 2002

[2] RFID für Dummies; Sweeney, Patrick;
Wiley-VCH Verlag; ISBN 3-52770263-6, 2006

[3] White Paper RFID – Technologie, Systeme und Anwendungen;
BITKOM www.bitkom.org; 10117 Berlin-Mitte, August 2005

[4] BITKOM RFID Guide 2006;
BITKOM www.bitkom.org; 10117 Berlin-Mitte, August 2005

[5] Technical Report EN 302 208-1;
European telecommunications Standards Institute ETSI, 2004

[6] Upstreaming RFID - Beyond Tags and Readers;
TATA Consultancy Services, 2005

[7] The Application Level Events (ALE) Specification V1.0;
EPCglobal Inc, www.epcglobalinc.org, 15-Sep-2005

[8] Basic Profile Version 1.0, WS-i Final Material; K. Ballinger, D. Ehnebuske, M. Gudgin, M. Nottingham, P. Yendluri, April 2004;
http://www.wsi.org/Profiles/BasicProfile-1.0-2004-04-16.html.