Collaborative Business Items

Christian Decker[1], Till Riedel[1], Michael Beigl[2], Luciana Moreira Sa de Souza[3], Patrik Spiess[3], Jens Müller[3], Stephan Haller[3]

1) TecO, University of Karlsruhe, Germany
2) DUS, TU Braunschweig, Germany
3) SAP Research CEC Karlsruhe, Germany

Keywords: CoBIs, Collaboration, Wireless sensor networks, Business Processes, Service-oriented Architecture

Abstract

Collaborative Business Items – or CoBIs – are an approach to build pervasive computing systems in order to support business processes with mobile assets and items. CoBIs enable the delegation of tasks originally located in stationary large back-end systems down to mobile assets and items. For instance, these tasks include the direct, i.e. item-centric, monitoring and detection of hazardous situations like they may occur when working with chemical goods. We refer to this delegation as *logic on the items*. The items are augmented by sensor technology and made aware of their surroundings and actions imposed on them. Collaboratively, they are able to reason on common states and enforce pre-defined conditions with only temporary support of stationary back-end computer systems. As a result CoBIs allow a more fine-grained, immediate, real-time and stable control of the handling of items. This paper presents the CoBIs fundamentals and results derived from implementations in a workplace safety scenario on handling various chemical goods. The results back up our claim that for the use of business logic in critical – e.g. dangerous – environments relocation of business logic onto items is required.

1 Introduction

Pervasive computing systems provide computing power and services virtually anytime and any-where in a surrounding to the user. This is achieved by embedding computation and sensing into real-world items and the provision of the results through communication. One well-known approach is the use of Radio-Frequency Identification (RFID) technology to track the identification and the location of items. The identification data together with the location of identification is collected by large back-end computer systems. At this stage business process logic is applied on the collected data set and appropriate actions are triggered. For instance, in a logistics scenario, the IDs of a group of items are acquired at the entrance of a facility, in order to check whether a delivery is complete. If it is successful, a forklift is notified, a storage location is assigned, and the supplier is notified and paid. If the delivery is incomplete, it is rejected. In this context, business logic is about modelling real life business items and their interaction with each other in various situations. The interaction of different business items and thus the logic are encoded in business rules and processes, which are executed in the back-end system. We propose that instead of only collecting information like with RFID, pervasive technology should interpret data and already execute part of the business logic, supplying the back-end with higher-level, pre-processed information. Executing the process logic in physical proximity of the data source decreases the amount of data that has to be processed by the back-end system. This results in less resource consumption for computation and communication and can in turn increase the responsiveness and the scalability of the whole system. In particular, we want to relocate parts of the tasks from the central back-end, where usually all business logic is deployed, down to an embedded pervasive technology-system embedded into items. By doing so, our approach enables business logic on the item, but also seamless connection to the back-end system in order to keep the coupling to the overall process. Figure 1 illustrates our approach.

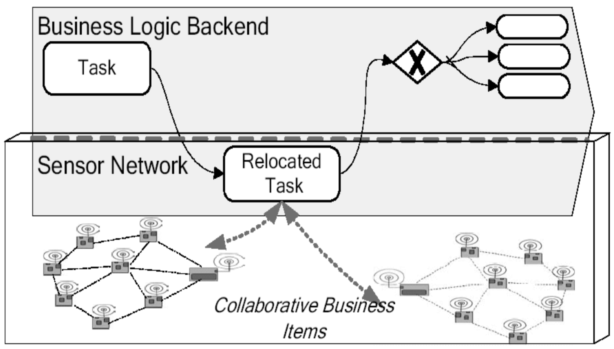

Figure 1. Relocation of business logic

Wireless sensor nodes are a preferable technology for our approach. Sensor node technology enables efficient data processing directly where information is originated. Sensor nodes can provide complex services to business applications by inter-node collaboration based on observation of their surroundings or by monitoring the items they are attached to. In the project CoBIs, partly funded by the EC, we investigate the approach of relocating tasks and business logic onto wireless sensor nodes. These sensor nodes

are attached or embedded into items, thus activating such passive items and making each item a active part of the business process.
The remainder is organized as follows: Section 2 analyses use and benefits of collaboration for pervasive computing systems. Section 3 presents the sensor node technology for CoBIs and section 4 reports on the coupling between sensor systems and enterprise systems. Section 5 presents the application of CoBIs in a real-world trial on workplace safety and quantifies results and achievements. Section 6 shows work related to the CoBIs approach, before section 7 concludes the paper.

2 Collaboration

CoBIs is focused on collaboration as the mechanism for processing diverse pervasive information. Collaboration, i.e. the cooperation of pervasive technologies in order to achieve a common goal, is therefore a key concept to build up efficient pervasive computing systems.

2.1 Collaboration to reduce load on back-end systems

The key motivation for CoBIs was mainly driven by the fact that back-end systems would experience a very high load when processing information from sensor-equipped environments. The immense amount of unfiltered information that can come from a steadily growing number of sensors can soon become a problem of scalability. The need for continuously evaluating sometimes-unreliable information often contradicts the goal of reliable, flexible and efficient processes. By processing the information directly on the embedded technology and in collaboration with the back-end system, high loads can be avoided, recognition granularity is improved and reaction time shortened. Long communication paths to the back-end and back to the environment are avoided, which allows real-time reaction processed within the physical environment.

2.2 Collaboration for efficient business logic processing

Wireless sensor nodes cannot replace a back-end system completely due to their limited amount of resources. Collaboration among the nodes may compensate those limitations by utilizing two key properties. The first one is distribution: While backend systems often need complex strategies to scale with an always-increasing number and speed of inputs, scaling is an intrinsic feature of sensor networks. This however does not mean that linear scaling is always sufficient to execute any algorithm. Especially communication can soon become a bottleneck. The second key property of most business logic is that it exposes a high locality concerning their information working set. As an example matching storage regulations against environmental conditions such as temperature or humidity can be done locally on a single node, as all input is available. At the same time a number of possible matching business rules can be statically evaluated on per-item basis, so that the logic actually executed on a single node only has linear time and space requirements. Practically, for local information processing only single hop communication is needed.

2.3 Technologies for implementing collaboration

Executing business logic on the items possesses a high locality of data processing, which can be efficiently exploited by sensor nodes. The wireless communication interface of sensor nodes often provides a notion of proximity. This can be further refined through technologies like ultrasound[4] or infrared[7]. Other key technologies for enabling collaboration focus on the description and runtime implementation of business logic, which are encoded as rules. For instance, programming languages for sensor nodes may be used to implement the distributed business rule evaluation. Conventional C programming can implement them efficiently, but such programs are hard to be updated in-situ, because they require the replacement of the complete sensor node firmware. When business rules change, e.g. because an item moves to a new location, an update is necessary. Other approaches, such as interpreted languages like Java for sensor nodes[8] are more flexible. They allow replacements of single classes during runtime. However, using imperative programming languages, collaboration has to be explicitly programmed as a distributed application.
A different approach is the arteFACT framework[10]. It consists of a Prolog interpreter, utilizing business rules in form of Horn clauses and proves for inconsistencies. Once it discovers one, an appropriate action may be fired. The advantage is that only some basic communication primitives have to be implemented. This type of collaboration strictly presumes that all information is available at the time of rule evaluation. The arteFACT framework was also used to implement a CoBIs scenario for workplace safety.

3 Wireless Sensor Network Technology

Pervasive computing environments rely on real-world interfaces – that is technology able to sense the environmental conditions and the processes or actions within the real world. Wireless sensor nodes and networks offer many different real world interfaces suitable for such environments. These devices integrate computing capabilities – primarily 8-bit micro-controllers – a wireless communication protocol stack and various sensors. Most of the device platforms follow a modular concept, where additional sensors can be added according to the needs of the application. Devices are battery powered and implement energy saving mechanisms to last for months or even years. In particular, for integration onto mobile items they are also very small.
CoBIs was implemented on the Particle Computer sensor node platform[1]. The node depicted in Figure 3 consists of the Particle Computer communication and processing board. It comprises an 8bit PIC18LF6720 microcontroller evaluating the business logic rules. Furthermore, it drives the RF front-end utilizing an RFM TR1001 transceiver on 868MHz with 125kbit/s. The sensor board, shown in the housing, comprises sensors for light, temperature, and

acceleration. Further analogue sensors can be added. Actuators are two ultra bright LEDs used for indication of business rule violation or regular operation directly locally to a human operator near the business item. For instance, the LEDs may indicate that immediate action is required to ensure workplace safety. Particle sensor nodes utilize the inherent proximity detection system. The adjustable sending field strength and the utilization of the receiver signal strength indicator (RSSI) allow a notion of proximity on the granularity of a less than 1 meter.

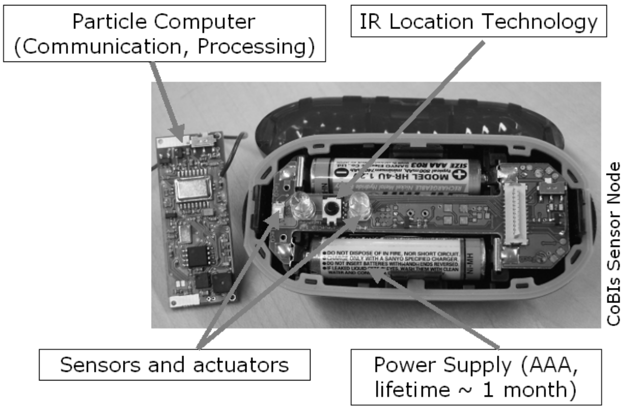

Figure 2. A CoBIs node based on Particle Computer

It is known that this information is not very stable or precise. Due to the intended usage of this technology in a workplace safety scenario, we require more precise location information. We additionally added an infrared (IR) location system for this reason. It registers the footprint of an adjustable IR beam emitted by a beaconI and provides a high accuracy regional borderline.,.

The Particle's ad-hoc communication protocol AwareCon[1] supports ad-hoc, mobile collaborative settings and real-time communication. The synchronization with a new network in range – the time until a new message can be sent to the neighbouring nodes – takes typically around 12ms. The mean delay for the synchronization with another single partner is around 40ms. The AwareCon protocol allows therefore the quick formation of a communication network between collaborating items. Furthermore, the protocol's non-destructive bit-wise arbitration mechanism allows very dense settings of nodes by guaranteeing a throughput only dependent on the number devices, which send at the same time. Usually, this number is lower than the number of devices in the network. Consequently, the communication performance remains high, even if the number of sensor devices scales up rapidly. These capabilities make the AwareCon protocol preferable for mobile and highly dynamic scenarios. The sensor node is embedded in a splash water resistant housing allowing an outdoor usage even under humid or rainy conditions.

4 Coupling Enterprise and Wireless Sensor Networks

While section 3 focused on one platform example, other platforms like Berkeley Motes[6], MITes[11], or µNodes from Ambient[5] are also useful for CoBIs usage. In this section, we give an overview on how to couple enterprise applications and wireless sensor networks.

From the perspective of an enterprise system, coupling with wireless sensor network raises the following questions: How are the systems *communicatively coupled* and how is business logic *modelled and distributed to the sensor nodes*? The broad heterogeneity of currently available hardware and communication protocols has led to many different solutions for both problems.

4.1 Service-oriented approach

We assume the software running on each network node is organized as a set of services. This means, that business logic is encapsulated and interfaces utilize a descriptive representation. Middleware components create so called service proxies, i.e. run-time representations of the embedded services that the enterprise system can access just like any other service. The service proxies can be used in different ways: 1.) Real-time data provided by a service proxy can be used to support the execution of one business process task. 2.) The proxies can be used to control which process task to execute next if more than one option is available. 3.) Parts of a business process can be relocated to the sensor system and the business process can be executed by using the corresponding service proxies directly. The first and second usages are well known concepts of integration of sensor systems into business processes, e.g. used in RFID tracking. We believe that the third option of relocating parts of the business process is novel and most beneficial for many use cases.

4.2 Technical implementation

The technical goal of the architecture designed for CoBIs is to enable the coupling of relocated process tasks provided by a heterogeneous software and hardware landscape. We also stress the seamless technical integration into existing service-oriented platforms.

The relocation of a process task starts with specific design and coding. During the design phase, it is necessary to define a set of services that collaboratively provide the functionality required by the business process. The main output of the design phase is a set of services , where for each service has a descriptive model defining functionality and interfaces by which the services are called.. We name this descriptive model CoBIs Language (CoBIL). CoBIL especially makes enterprise systems aware of the capabilities of sensor systems. For instance, the notion of a limited energy resource is usually not considered in enterprise applications. With CoBIL, enterprise systems are informed on the constraints and actively use them for managing relocated services. The result of the implementation phase is an executable program code that can run either on different platforms or on a specific platform. The code implements the business logic and is stored in the Service Repository, where it can be queried by via pointers within a CoBIL document.

The architecture in Figure 3 depicts mechanisms for managing the relocated processes. The *CoBIs Service Interface* provides the functionality described in CoBIL to business applications as web services. When the application makes an invocation, the *CoBIs Service Interface* forwards the request to the *Service Invoker*, which uses the *Message Handler* component to convert the invocation into the sensor system vendor-specific protocol. Subscription and event notification are handled by the *Notification Broker*. It distributes the event among the subscribed business applications. Both, Message Handler and Notification Broker, utilize a generic mechanism of protocol transformation for conversion of XML based SOAP message calls to tuple-oriented wireless sensor network protocols [2].

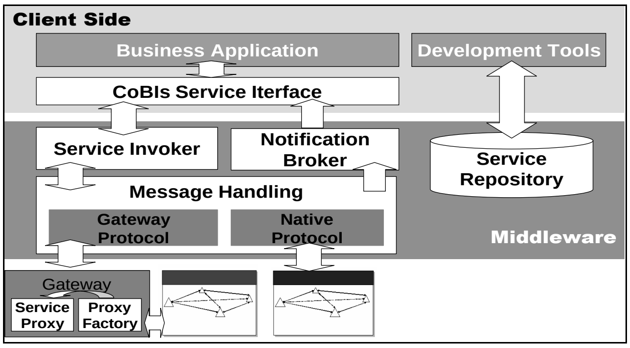

Figure 3. CoBIs Architecture

The message handler may exploit specific gateways or native protocols adaptors. While the gateways hidethe platform-specific complexity, the native handlers are usually more complex but also more efficient due to lower overhead and closer coupling to the application. However, native handlers require deep insight in specific protocols, more effort for the implementation and are less flexible. We investigated the coupling of the systems in more detail in [2] along with some experiences and guidelines for developers. A concrete implementation example utilizing UPnP as service gateway middleware is analysed in [9].

5 Implementation Scenario

We implemented and evaluated the collaborative business items in a workplace safety scenario at BP's largest acetyls production site for acetyls in Hull, UK. Two trials were conducted in a declassified storage area on June 2006 and November/December 2006. In each trial 20 chemical drums were equipped with CoBIs technology based on Particle Computers as depicted in Figure 4 were installed for about one month without any further local supervision by a human operator. System management tools and utilities periodically and remotely reported on events happening on the site.


Figure 4. Chemical drums with Particle Computers

5.1 Setup

The CoBIs scenario was set up in two different locations (GS12 and GS16 storage house[1]) on the site. Both locations stored 20 chemical drums. Every location was equipped with a gateway router that forwarded packets emitted by Particle nodes to the local area network provided by BP. Each gateway router was also connected to an infrared beacon that sent out location information that could be received by the nodes in order to determine their current location. The setup defined three different locations, two in store house GS12 and one in GS16. The two gateway routers in GS12 were connected to the local area network by a WiFi link. The third gateway router in GS16 was connected directly to the local are network. The Figure 5 illustrates the network setup for the implementation scenario.

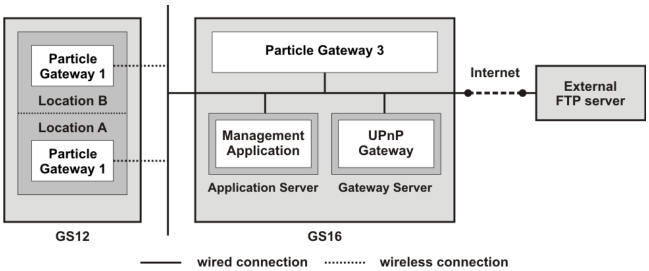

Figure 5. Network setup of the implementation scenario

The business logic for the storage regulations was modeled in SAP's EH&S (Environment, Health and Safety) system. The system was used to parameterize a hazardous goods detection implemented on nodes. It needed location information input from a location beacon system. Furthermore, the nodes monitored the temperature in order to check environmental storage regulations. Additional information (voltage, duty cycle) was provided to manage the networks functionality. The gateway software to connect the sensor network to configuration and logging utilities was run on two 200 MHz embedded Linux MIPS systems in combination with a with 2GHz Intel Windows XP server running the SAP application server. The SAP

[1] The identifiers result from a BP internal safety code.

server and the sensor network were coupled using UPnP. This technology was chosen to provide a standardized interface of proprietary sensor node to back-end systems. The gateways and the application server were only installed for remote logging and configuration purposes. They did not affect the collaboration mechanisms and distributed business rules evaluation on the sensor node technology.
Setup time for installing all components was about one week. Afterwards, the system was in operation for one month. Periodically, recorded data was uploaded to an ftp server in the Internet allowing to check and evaluate the current operation status.

5.2 Goals

The goal was to model storage regulation of chemical substances stored in two different stores with multiple storage locations inside the same store. We implemented the services for the business logic for supervision of multiple types of storage regulation:

- Storage limit per chemical per location
- Incompatibility classes of chemicals stored in the same location
- Environmental constraints (maximum/ minimum temperature)
- Maximum time in storage

Further, we wanted to evaluate the performance of the overall CoBIs system. In particular, we were interested on the effects of collaboration as analyzed in section 2.

5.3 Location

Crucial for the storage regulation is the determination of the current location. Particle nodes receive location information from infrared beacons that were placed in each logical storage area. The beacon generates a diffuse infrared signal encoding a pre-defined identification number (location ID). The strength of the signal was adjustable. After a calibration phase the location system operated reliably on a per-shelf granularity. Using diffused infrared the signal is not dependent on a direct line of sight but may weaken its high accuracy in terms of detection area. Another disadvantage could be possible IR saturation by direct sun light. We have not experienced these issues as a limitation, because the chemical drums were stored inside and by default positioned sufficiently far to avoid overlapping infrared footprints.
In the second trial we performed a test where a forklift moved a whole pallet with water drums to another storage area with incompatible goods (see Figure 6).

Figure 6. Fork lifter with drums

The nodes ran the location detection at a speed of 40Hz. Therefore, we could also run stress tests for location misdetections and evaluate the accuracy of the infrared location system. The nodes collected the location IDs during the transport and reported to the back-end system immediately. The figure below depicts the results recorded from 3 Particles. Fluctuations may occur, but remain only for a very few seconds. Once placed in a location, the location detection is reliable.

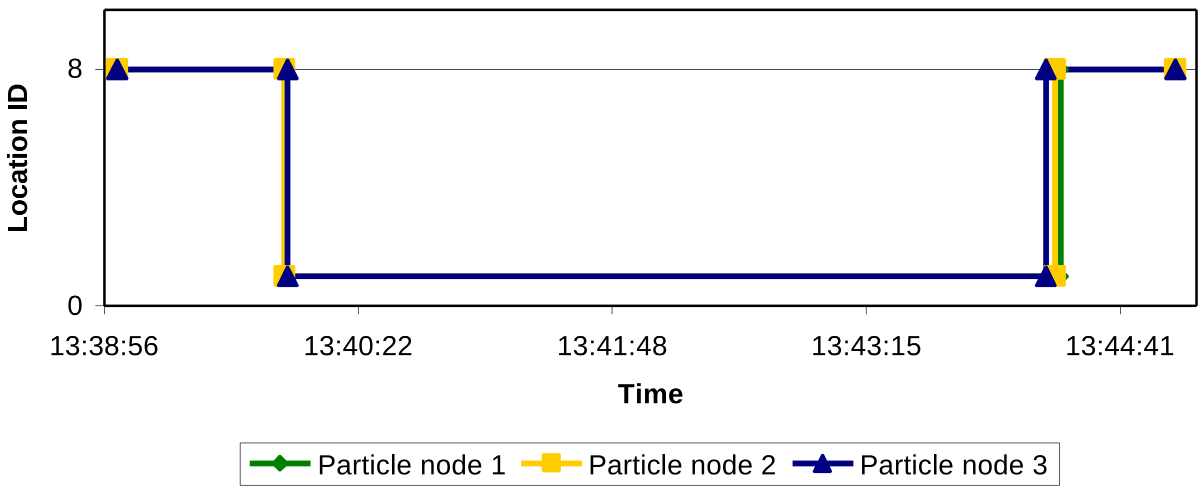

Figure 7. Location tracking

The senor node's resources needed for determining location are a TSOP infrared receiver. The location service requires 746 byte of program flash and a low overhead of 10 byte of RAM on the micro-controller. The code resources are primarily used for the implementation of a low-pass filter in order to filter out noise on the infrared channel.

5.4 Collaborative Detection of Hazardous Situations

Hazardous situations occur if storage regulations are violated. The detection of these rules is implemented in the hazardous goods service on the Particle sensor nodes. The hazardous goods service is loosely coupled to the location service and takes its input to compute storage incompatibilities and limits. Limits and incompatibility classes can be configured via the service interface to the back-end system. The services communicate via a tuple based interface that sends out information about drum size, stored chemical class and location to the other nodes on a location change as well as periodically. The table below lists the sampling rates for the services participating in the hazardous situation detection.

Service name	Sampling rate
Hazardous Goods Service	1 Hz
Sensor Service	Temperature: 0.05Hz Voltage: 0.4Hz
Location Service	IR Location Sensor: 40Hz

Table 1. Sampling rates of the services on a sensor node

By collaboratively updating the storage state of each service instance, the regulations can be checked locally on each node. Local alarms are then also signalled collaboratively within a location. The hazardous situation due to storage violations was recognized within 1 second. According to BP up to 10 seconds are acceptable. In our experimental setting, flashing of the bright LEDs indicated the hazards. Clearly, the advantage is in-situ, real-time reaction by shortening the communication path and let the items directly reason on the situation without an active involvement of the enterprise system .

The business logic for the hazardous goods service on the Particle sensor node consumes 8742 bytes of program flash and 242 byte of RAM. Additionally to the location service it depends on a temperature service for monitoring according minimum and maximum regulations and as well as network services for communication.

5.5 Performance

The effects of collaboration within CoBIs for business systems and in particular for pervasive computing applications are analyzed in section 2. This section presents quantitative results from the implementation scenario.

We first looked on the communication load. We compared the expected load, which we have drawn from assumptions on the scenario, e.g. no message loss, with the actual measured values. The observations shown in Table 2 crosscut the complete CoBIs architecture from the business application level down to the wireless sensor network. Most messages, were exchanged for logging and not for use by an application. This explains the high rate of messages filtered out by the gateway.

Parameter	Expected value	Actual value
Average message load on the application server level	20 msg/min	7 msg/min
Peak message load on the application server level (all drums put to same location)	400-500 msg/min	212 msg/min
Average message load from the sensor network	N/A	187 msg/min
Comparison: messages on gateway level vs. messages on application server lev.	80% of the messages are filtered out	96% filtered out
Propagation of new rules/configurations	20s per drum	23.3s per drum (210s for nine drums)

Table 2. Communication load (msg/s = messages per second)

We also tested the update of the business logic and transfer of new configuration such as storage limit changes. The propagation time is mainly caused by a communication duty cycle (current implementation: 60%) of the senor nodes. An adaptive scheme, powering up the nodes for a longer time could further reduce the delay.

Crucial for the performance of the overall system is the performance on the gateway level. Relocated business logic utilizing collaboration among the sensor nodes keeps information local within the sensor network. As a result, the number of messages forwarded to the application can be significantly reduced, which consequently reduces the load on the enterprise system. Since we are logging the complete traffic within the sensor network, we express this reduction as a filter at the gateway. The performance results can be seen in Table 3. Collaboration reduces the message load in case of hazardous detection events from 10 messages/s down to 2 messages/s. For every service on the sensor node the gateway instantiates one service proxy. Each node runs four services (see Table 1).

Parameter	Actual value
Number of nodes	>= 20
Number of services/proxies	>= 80
Avg. Gateway event load prior filtering/service transformation	10 msg/s
Avg. Gateway event load after filtering/service transformation	2 msg/s
Processing time for transformation	< 100ms
Processing time for (non-blocking) invocation	< 10ms

Table 3. Performance results of the gateway (msg/s = messages per second)

The coupling of sensor networks with enterprise system requires protocol transformations (see section 4). The transformation delays were acceptable for the implementation scenario.

5.6 Achievements

The logic pushed to the sensor network worked reliable. Both the business software running on an SAP application server as well as the sensor network performed their services according to their specification. The average message load to the business system was only about 30 messages per minute, mostly resulting from voltage monitoring need for a system management application in the back-end.

Surprisingly most of the problems encountered on site were related to technologies not specific to the collaborative business items. BP provided us with a wireless 802.11 based infrastructure we were able to connect our gateways without the need of complicated wiring on site. Our lab experiments showed that bandwidth was no issue at a setting of this size. A lab installation of our hardware on top of an 802.11 network also indicated no problems. Some problems however arose from some details of the actual setting. Probably because of the rather humid weather conditions close to the cooling towers the 802.11 network showed a high packet loss at times. This packet loss did not very much affect TCP traffic but the unacknowledged traffic like UDP. UPnP uses UDP multicast as part of its discovery protocol, just as the DHCP based dynamic addressing for the gateways. As a result, the back-end system experienced long discovery delays and a delayed notification of events. Executing business logic on the items made the reactions of the sensor network subsystem very robust, especially in critical situations. Our experiences therefore greatly support our initial assumptions that business logic has to be executed on the items to provide enough stability for extreme – e.g. dangerous – settings.

6 Related Work

A similar safety scenario compared to the CoBIs one was ChemSecure conducted by the NASA's Dryden Flight Research Center [12]. NASA did use passive RFID tags attached to chemical containers. Each RFID tag had a unique identifier that was listed in a database. The environment such as facility doors and trucks were equipped with RFID readers and monitored the containers. The database system in the background constantly cross-checked data from the RFID tags in order to match them against possible conflicts. Hazards were notified to responsible persons via e-mail or short messages on cell phones. The CoBIs approach runs in a distributed manner. As a result, the devices are more complex than passive RFID tags. This allows CoBIs to notify in-situ where the worker is still around. Furthermore, it requires significantly less infrastructure and gains more flexibility.

The authors in [3] sketch out a CoBIs like scenario for transport logistics. An application case considers returnable transport items such as palettes. The goods on the transport containers should comply with certain transport conditions such as environmental conditions, correct loading truck, correct position in the cross dock, etc. The paper discusses various requirements on a potential rule engine, localization mechanisms, and communication protocols. Only components and a solution outline are given. It is definitely worth to offer the current CoBIs implementation as an example technology for the logistics scenario.

7 Conclusion and Outlook

With collaborating business items (CoBIs) we have shown a concept of information processing in ambient intelligence environments. Instead of processing on stationary back-end systems, the acquired environmental data is directly processed on the item according to a so-called business logic encoded as rules. We refer to this approach also as *logic on the item*. Collaboration is one of the key concepts here in order to enable efficient local and in-situ processing. The implementation of the CoBIs approach utilizing modern wireless sensor network technology and reported results and experiences in real-world business case within a workplace safety scenario have strongly supported the benefits of this approach. In spite of the still prototypical nature this system we hope that our experiences can help the deployment and integration of sensor node technology into live business applications in the near future.

Acknowledgements

The work presented in this paper was partly funded by the European Community through the project CoBIs (Collaborative Business Items) under contract no. 4270. We further like to thank Paul McCune, Mike Haley and the people at BP Chemicals Saltend for their support during the application trials.

References

[1] C. Decker, A. Krohn, M. Beigl, T. Zimmer. The Particle Computer System. Proceedings of the ACM/IEEE Fourth International Conference on Information Processing in Sensor Networks, Los Angeles, 2005

[2] C.Decker, P.Spiess, L.M. sa de Souza, M.Beigl, Z.Nochta Coupling Enterprise Systems with Wireless Sensor Nodes: Analysis, Implementation, Experiences and Guidelines. PTA Workshop on Pervasive 2006, Dublin, Ireland, May 2006

[3] L.Evers, M.J.J. Bijl, M. Marin-Perianu, R.Marin-Perianu, P.J.M.Havinga. Wireless Sensor Networks and Beyond: A case study on transport and logistics. International Workshop on Wireless Ad-Hoc Networks (IWWAN 2005), London, Uk, May 2005.

[4] M.Hazas, C.Kray, H.Gellersen, H.Agbota, G.Kortuem, A.Krohn. A Relative Positioning System for Co-Located Mobile Devices. 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys) 2005, June 6.8, Seattle, USA.

[5] P.J.M. Havinga, The Quest for Low Cost, Ultra Low Power Wireless Networks for smart environments, White Paper from Ambient Systems, January 2006

[6] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, K. Pister, System architecture directions for network sensors, ASPLOS 2000, Cambridge,November 2000.

[7] A.Krohn, M.Beigl, M.Hazas, H.Gellersen, A.Schmidt, Using Fine-Grained Infrared Positioning to Support the Surface-Based Activities of Mobile Users. 5th International Workshop on Smart Appliances and Wearable Computing (IWSAWC) 2005, Columbus, USA

[8] T.Riedel, A.Arnold. An OO Approach to sensor programming, European conference on Wireless Sensor Networks (EWSN) 2007, Poster Abstract

[9] T.Riedel, C.Decker, P.Scholl, A.Krohn, M.Beigl. An Architecture for Collaborative Business Items. ARCS 2007, Zurich, Switzerland

[10] M.Strohbach, H.-W.Gellersen, G.Kortuem, C.Kray. Cooperative Artefacts: Assessing Real World Situations with Embedded Technology. Ubicomp 2004

[11] E. M. Tapia, N. Marmasse, S. S. Intille, and K. Larson, MITes: Wireless portable sensors for studying behavior in Proceedings of Extended Abstracts Ubicomp 2004: Ubiquitous Computing, 2004.

[12] L. Sullivan RFID Lets NASA Monitor Hazardous Materials, Information Week, January 10, 2005.