

Specific Programme
Integrating and strengthening the ERA

Project Title
Collaborative Business Items

Project Acronym
CoBIs

Project No.
IST-004270

Deliverable 401
Management & Maintenance Requirements

Work Package 4
System Management

Deliverable Leading Partner: Infineon Technologies

Authorization Level: PU

January, 2005

Version 1.0

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Document Information	Deliverable	D401

Document Information

IST Project Number	FP6 – 004270	Acronym	CoBIs
Full title	Collaborative Business Items		
Project URL	http://www.cobis-online.de		
EU Project officer	Rolf Riemenschneider		

Deliverable	Number	401	Title	Management & Maintenance Requirements
Work package	Number	4	Title	System Management

Date of delivery	Contractual	01/2005	Actual	01/2005
Status	final		final	<input checked="" type="checkbox"/>
Nature	Prototype <input type="checkbox"/> Report <input checked="" type="checkbox"/> Dissemination <input type="checkbox"/>			
Dissemination Level	Public <input checked="" type="checkbox"/> Consortium <input type="checkbox"/>			

Authors (Partner)	P. Spieß (SAP), C. Decker (TecO), K. Douglas (BP), B. Haberler (Infineon)		
Responsible Author	B. Haberler		Email berthold.haberler@infineon.com
	Partner	Infineon	Phone +43 316 7271 317

Abstract		
Keywords		
Version log/Date	Change	Author

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Table of Contents	Deliverable	D401

Table of Contents

Document Information	2
Table of Contents	3
List of Figures	4
1 Executive Summary	5
2 Introduction	6
2.1 Relation to other Work Packages	6
3 Short Scenario Analysis and End User Requirements	8
3.1 CoBIs Scenario “Smart Shelf”	8
3.2 CoBIs Scenario “Chemical Drums”	9
4 Technical Requirements from a Service Oriented Perspective	12
4.1 Comfortable Modelling of Business Logic	12
4.2 Monitoring Network State before Deploying.....	15
4.3 Mapping Services to Nodes.....	16
4.4 Injection and Start of a Service	17
4.5 Monitoring a Running System	17
4.6 Service Remapping and Auto-Deployment during Operation	18
4.7 Configuring a Running System	19
4.8 Updating a System	20
4.9 Licensing and Security	20
5 Conclusion and Future Work	22

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	List of Figures	Deliverable	D401

List of Figures

Figure 1 - Parts of the System Influenced by WP4	6
Figure 2 - CoBIs nodes enable Smart Shelf Systems.....	8
Figure 4 - Steps of Deploying a Service	12
Figure 5 - Service Hierarchy and System Management.....	13
Figure 6 - Re-mapping of Services during Operation.....	19
Figure 7 - A Service in the Service Repository.....	19

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Executive Summary	Deliverable	D401

1 Executive Summary

The CoBIs WP4 objective is to develop off-line and run-time support for modelling, monitoring, deploying, configuring, and updating services in a large scale CoBIs networks. The expected result is the design and implementation of dedicated administration services.

The report surveys the management and maintenance requirements of CoBIs systems by incorporating the specifications of dedicated use cases. A hierarchical and systematic approach for a service based administration platform is presented. The achieved results allow accommodating the upcoming implementation steps within the overall service architecture developed by the cooperating CoBIs work groups.

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Introduction	Deliverable	D401

2 Introduction

The establishment of CoBIs enabled environments involves the development of appropriate management and maintenance concepts and tools. Work package 4 incorporates the requirements from the end user as well as the technical concepts and tools for maintenance tasks.

CoBIs networks will be complex systems fulfilling complex tasks. However, ease of use will be a key element for the future success of CoBIs. Users have to be empowered in a way that they can monitor the state of the network at any time. Business professionals must be able to specify the business logic in a high-level, descriptive and hardware-independent manner. Deployment and technical decisions should be automated by intelligent software services. Technology experts must get comprehensive internal data to fine tune the system or to detect errors.

Because the management and maintenance requirements primarily arise from the application-side, this report starts with a short presentation of the concrete CoBIs technology use cases. The functional features of the proposed management system are derived by looking at the requirements from the end-user side. A hierarchical and systematic approach for a service based administration platform is presented in the following. The derived catalogue of abstract system requirements summarizes the services which have to be implemented, in order to manage and maintain large scale CoBIs systems in real-world applications. This report ends with a conclusion and a brief overview of future work.

2.1 Relation to other Work Packages

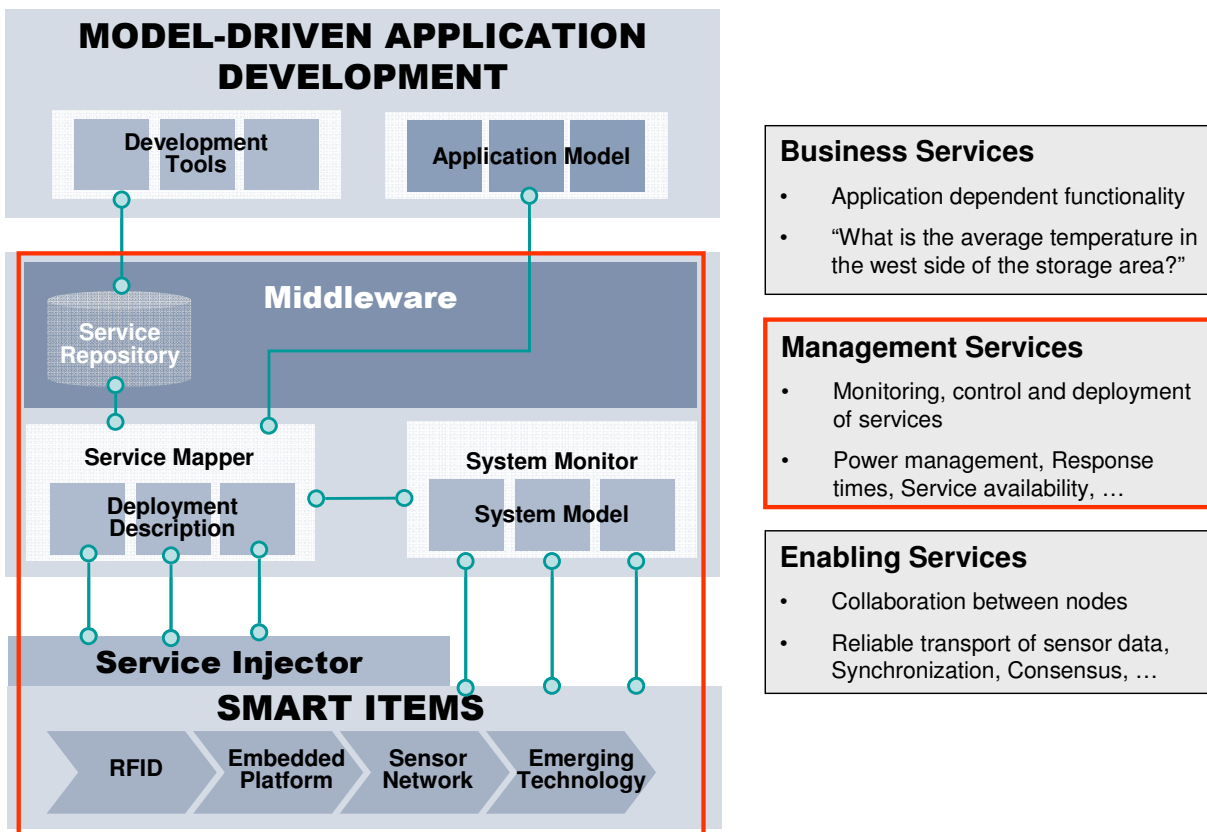


Figure 1 - Parts of the System Influenced by WP4

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Introduction	Deliverable	D401

WP4 is concerned with System Management, a task that influences nearly all parts of a CoBIs system as well as the system architecture itself. Figure 1 shows the parts that are particularly influenced by WP4.

It is dependant on application requirements from WP5 and hardware constraints delivered by WP3. WP1 provides input in the form of a sketch of the needed functionality and finally WP2 bridges the gap between hardware functionality and business needs. WP4 seeks to make all the parts of the system work together in a flexible way and provide the “glue” and functionality make the system work, keep it working, and ensure that it runs within a given set of parameters.

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Short Scenario Analysis and End User Requirements	Deliverable	D401

3 Short Scenario Analysis and End User Requirements

The CoBIs research results will be deployed in real world application trials, in order to validate the developed frameworks. The application scenarios can only be supported by introducing collaborative, networked, embedded technology into the business processes. Feedback from the implementation of CoBIs technology in real world processes will become available through the application trials at BP and Infineon. The BP scenarios are mainly focused on safety-critical processes as a particularly challenging business application domain, while the Infineon scenario deals with distributed processes for wireless networked RFID readers.

In the following a short introduction to exemplary CoBIs technology use cases is presented, in order to justify some of the maintenance & management requirements of CoBIs systems. A more detailed description of specific application trials is given in D501.

3.1 CoBIs Scenario “Smart Shelf”

The idea to make items smart by attaching a transponder holding a unique identifier opens a wide range of new applications for logistic processes. Tremendous quality improvements and cost savings in the logistic flows are expected by fully automated inventory systems. RFID is a next generation tagging solution that applies to various aspects of the supply chain. In retail settings, the chips or "smart tags" replace the barcodes found on packages today, and can provide many other supply chain benefits. Today’s RFID logistic solutions provide real-time inventory information to software that can measure sales campaigns and trigger replenishment orders. The direct access to stock and throughput data allow to establish new store management procedures, according to the needs of retail enterprises and their customers.

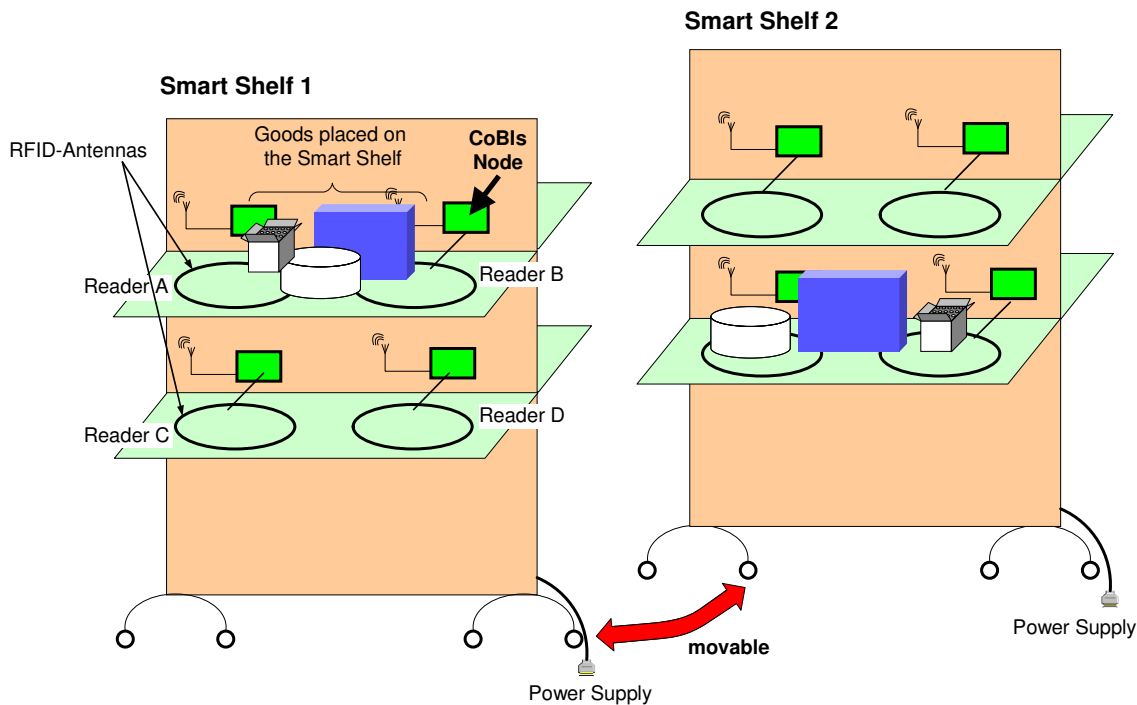


Figure 2 - CoBIs nodes enable Smart Shelf Systems

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Short Scenario Analysis and End User Requirements	Deliverable	D401

Figure 2 shows an Intelligent Warehouse Shelf, which makes use of machine "readers" that track nearby items in a supply chain. The so-called Smart Shelf is a new RFID enabled logistic solution which is currently under development.

In use, Smart Shelves could take stock of the goods lying on them and report incoming or outgoing goods to the IT infrastructure. Imagine to enter a store and to find out easily at an information terminal, if clothes of fitting size and preferred colour are available and where you can find them. Besides this customer-related application scenario, there are many other new logistic features that would be enabled by implementing a Smart Shelf System:

- By comparing data from shelves with data generated at point of sales, measures against theft could be defined.
- The system could drive smart price labels. If goods are moved within shelves, or to other shelves, the displayed pricing information would follow automatically.
- Smart displays with integrated Sindrion Transceivers could be easily added to the system for advertising and information purposes.
- Smart temperature sensors could be implemented into the packaging of goods and into freezer compartments in order to permanently control the quality of frozen food and autonomously trigger alarms.
- As moving shelves and changes in the arrangements of furniture have to be considered, a wireless communication ensures convenient handling of the system.

In supermarkets and warehouses hundreds of shelves might be required to set up a comprehensive solution. Because communication between readers, displays, sensors, and the IT infrastructure will be performed by a low cost wireless link, a major challenge for the project is to overcome the problem of RFID reader interference. Radiated fields of neighbouring RFID antennas influence each other in a way that communication with transponders is prevented. Therefore neighbouring readers have to be switched on and off so that interference is prohibited. As moving shelves and changes in the arrangements of furniture have to be considered, a fully automated interference check service has to be applied, so that readers can mutually agree to a time multiplex scheme. Many of the proposed system features have to be performed fully autonomously.

An easy to install, flexible, easily maintainable, and cost effective architecture is necessary to justify the proposed business case. In order to construct a network of Intelligent Shelves, the technology architecture of the smart shelf has to be an embedded system equipped with several readers acting as collaborative business items.

Feedback about the CoBIs technology will become available through a Smart Shelf demonstrator, which will be set up at Infineon's RFID Solution Excellence Centre in Graz, Austria, and at an industrial selling point of Gardeur AG, a producer in the textile industry located in Düsseldorf, Germany.

3.2 CoBIs Scenario "Chemical Drums"

This is a specific case of a more general scenario for potentially hazardous items to be self aware and generate safety alerts automatically. At this stage this functionality would be to augment

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Short Scenario Analysis and End User Requirements	Deliverable	D401

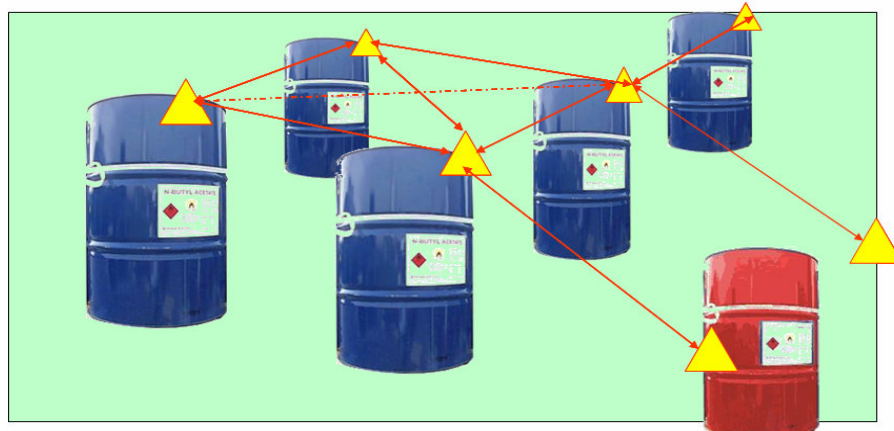


Figure 3 – CoBIs nodes in the “Chemical Drums” scenario

manual/human processes and not replace them. In the general case “drums” of chemicals range in size from 1 to 50 litres (although larger containers should be accommodated) would be equipped with CoBIs nodes and “programmed” with information on their content as well as the HSE¹ logic described in the detailed test cases below. This information should contain data about the chemical and how to handle spills etc, but this functionality does not form a critical part of the test. The physical characteristics of the tags are not defined – however there may be a requirement for intrinsically safe devices for live site tests. This should be flagged as a potential issue

This trial would be carried out on site at Hull. No specific chemicals storage area is defined at this stage as they change frequently. A user with a handheld reading device or an automatic portal should be able to interrogate each drum manually and obtain information on its contents, HSE rules, and recent processing history.

Test case 1: Safe limits for local storage exceeded

Safe limits for volume of chemicals are defined in terms of total volume, each drum must therefore be aware of its own volume and of the total safe, local limit. If the limit is exceeded then an alert should be raised (locally) or via a gateway. Specific test: An arbitrary limit of 200 litres of peroxide will be established. An alert should be generated if and only if that limit is exceed for drums stored in any configuration in a room space 10m x 10m. Drum sizes will vary between 1 and 50 litres. Maximum number of peroxide drums will be 40. Maximum number of other drums (containing other chemicals) will be 60.

Test case 2: Incompatible chemicals in proximity

It is unlikely to be feasible to list every chemical with which another is particularly reactive and so each drum/tag must be aware of the “class” and “hazard rating” of its contents. BP will provide lists of these later in 2004. Determining proximity will be a potential issue. Unsafe proximity may be described less than a metre or over 20 metres.

¹ HSE is an abbreviation for Health and Safety Executive, the British government body responsible for the regulation of risks to health and safety in the UK. CoBIs seeks to automatically assure conformity with HSE rules.

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Short Scenario Analysis and End User Requirements	Deliverable	D401

Specific test: Same test area as test case 1. Any incidence of incompatible chemicals being stored in this space should generate an alert. It will be acceptable to have two forms of alert. One = “possibility” of proximity. Two = “definite breach of proximity”

Test case 3: Invalid storage area

Chemicals requiring storage in special areas should be aware of this. BP will define up to six different storage requirements (e.g. temperature controlled explosion proof etc.) More than one storage requirement may apply at a time for any given drum. The storage area itself must be able to “identify itself” to the drums. No requirements for this are defined.

Specific test: After an arbitrary time out period an alert should be generated if a drum/tag cannot identify that it is stored in a safe area.

Test case 4: Expiry or inspection date exceeded

This is perhaps the most straightforward case. Each drum/tag should be aware of both inspection and expiry/usage dates and generate 1) a warning when nearing that date (arbitrary time window), 2) an alert when that date is exceeded. Note that there is a special case of this scenario where reusable drums are used. The drum itself will have an inspection date and this is different from that of the contents.

The complete description of the “Chemical Drums” scenario, quantifying real world parameters, etc., is maintained in deliverable D501.

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Technical Requirements from a Service Oriented	Deliverable	D401

4 Technical Requirements from a Service Oriented Perspective

This chapter will specify the requirements of the administrative functions of CoBIs from a user's point of view. To assure that the list of requirements is complete, we will go through all steps from designing the high-level services a CoBIs network should provide, via the deployment of these services to the execution, monitoring and failure detection of the running services. At each step we will note the requirements to the system.

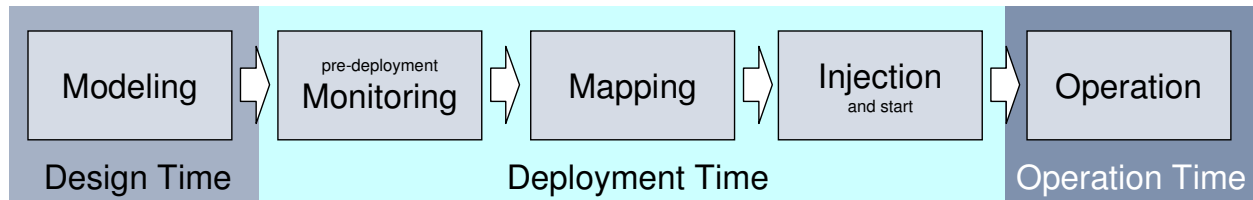


Figure 4 - Steps of Deploying a Service

4.1 Comfortable Modelling of Business Logic

When composing a service, the designer has a special business task in mind that the hardware of the wireless sensor network should fulfil. Since the designer of a CoBIs network is often not a trained computer specialist but rather a business professional, we can't expect him or her to write code for a wireless sensor network platform - be it for some virtual machine, for an operating system or directly for various microcontrollers of a sensor node.

To give the user a powerful means for specifying business logic and at the same time to free him from the complexity of designing and implementing complex distributed algorithms, we suggest an approach of graphical modelling. This approach implies the existence of some components that can be "wired" together to fulfil a higher goal, and for even higher flexibility and seamless enterprise software integration we want to use a service paradigm.

From a management point of view, we introduce tree classes of services: *Device services* that originate from the usual capabilities of wireless sensor network nodes, e.g. detecting other nodes in a certain range, sending data, subscribing to data coming from other nodes, reading sensor values etc. These very low-level services encapsulate the functionality of a device. The device services will be contributed by the project partners involved in WP3.

On top of the device services we introduce *enabling services*. The enabling services build a hardware-independent but still very technical abstraction of the device services. Examples for enabling services are sending a message to all direct neighbours or to the backend, or reading a sensor value and transforming it in a useful scale - all in a hardware-independent way. Those services form a platform on which applications are built as well as for service deployment and operation. They provide general, application-independent functionality and will be mainly provided by the WP2 contributors.

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Technical Requirements from a Service Oriented	Deliverable	D401

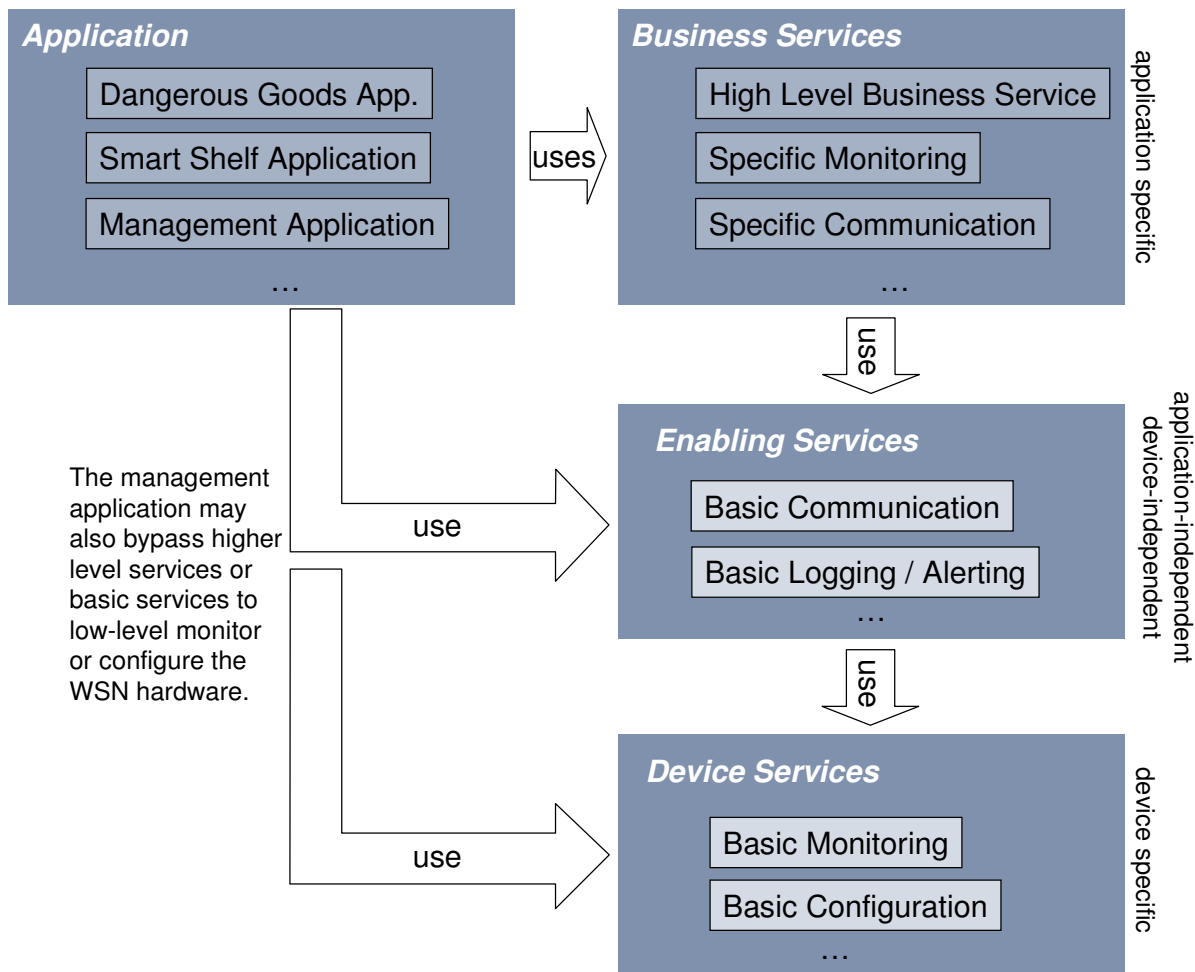


Figure 5 - Service Hierarchy and System Management

The set of enabling services has to be carefully selected. Too few of those services would either lead to services with a very complex interface or to missing functionality. On the other hand, having too many enabling services would cause the foundation for service composition very scattered and since all of those services had to be implemented, they could exceed designated project development time of the project partners or hardware memory limitations of the sensor hardware.

We expect an off-the-shelf CoBIs node to be pre-programmed with both device and enabling services so that they are available after turning it on for the first time.

The third tier of services comprises higher level, mainly application-specific compound services. These services will be defined by WP1. Figure 5 depicts the hierarchy of services.

In order to compose services by combining other services, there also has to be a *descriptive model* of a service's interface. The model has to specify which data a service offers or accepts, and whether communication is synchronous or asynchronous.

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Technical Requirements from a Service Oriented	Deliverable	D401

To finally combine existing services into compound services, we need a model for that process. The model of a compound service must include the services it uses as well as the way they are combined, e.g. how messages flow between the used services.

Optionally, an experienced human modeller with technical insight should also be able to specify technical requirements of a service in a *technical requirements record*, a part of the CoBIL document assigned to the service. It could include minimum required bandwidth or message failure rate. Another example would be to indicate that the service is likely to consume lots of battery power. This information can help during the deployment phase as it can be used to optimize the process of assigning services to network nodes.

After a service has been designed, either by composition of other services or implementation in code and the service's interface and technical requirements have been described in a machine-readable CoBIL document, both are submitted into a central service repository.

CoBIL will be an XML-Based service description document. Every service available in a CoBIs system must have a CoBIL document assigned to it.

It is up to work package 1 to describe service modelling in detail as well as identifying the enabling services a CoBIs network has to provide. Work package 4 needs input from WP1 here.

The service approach is very desirable because it allows seamless integration in today's business software, which is build more and more in a service-oriented way. Services have the advantage of providing information throughout the enterprise in a platform and language independent manner.

However, the expertise of our project partners also allows us to investigate in a second direction as far as modelling of business logic is concerned: University of Lancaster has also built a system based on horn clauses, known from Prolog systems. That system has been successfully used for first experiments on the application scenario from bp plc. under lab conditions. Such sets of horn clauses could also be modelled graphically.

Having the broadest install base in today's enterprises, we want the modelling software (as well as all other PC software developed in this project) to run at least on Microsoft's Windows-NT based systems Windows 2000, XP and Server 2003.

Identified Requirements:

- Service description model (description in machine-readable CoBIL)
- Service composition model for compound services (CoBIL)
- Services can have a *technical requirements record* (CoBIL)
- Identification, selection and implementation of enabling services on the nodes (description and implementation)
- Graphic modelling tool for composition of higher-level services by combining existing services
- A central repository where service descriptions and service implementations are stored.
- As an additional approach: Graphic modelling of horn clauses and their distributed resolution

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Technical Requirements from a Service Oriented	Deliverable	D401

- All management software should run at least on Windows NT-based systems from Microsoft

4.2 Monitoring Network State before Deploying

A CoBIs network must constantly be monitored. In this section we will point out the usefulness of monitoring prior to service deployment.

Before we start, we imply that the CoBIs hardware has already been attached to the business items we want to monitor. Once the “blank” nodes are attached to the items that should be monitored and activated, they execute a set of device services that have been pre-programmed on each node. The device services provide remote programming and access functionality as well as basic monitoring of low-level parameters of the network.

It is strongly preferable that also the enabling services defined during the project (or at least a reasonable subset of them) are running on a “blank” node. This has the advantage that programming and communication may be approached in a device and platform independent way. On the other hand, accessing device-specific services allows for deeper optimization of tasks.

For monitoring, a reactive as well as a proactive approach is practicable. These approaches only represent extreme points. When designing real systems you always choose a suitable compromise between them.

Reactive monitoring means that the monitoring task stays in the background and passively watches the state of a node and communication of the services on it, making decisions based on the observed data. The advantage is that no additional communication overhead is used for monitoring. On the other side, this kind of monitoring can fail to detect errors like link breakage or report an error condition with long delay.

The option on the other extreme is proactive monitoring that consumes additional bandwidth, memory and computing resources in order to detect unfavourable situations more reliable and faster.

Communication is the most energy-consuming task on a WSN node, and dividing communication into sending and receiving, sending has higher energy costs than receiving. This leads to the requirement that, for the sake of a long battery life, monitoring should be kept as reactive as possible, e.g. by doing self tests on the hard and software of a node, and should predominantly rely on the overhearing of regular inter-service traffic. However, when a potential erroneous situation is detected or the user or some service in the system requests more detailed information about a certain property of the network, that property should be actively inspected.

For post-analysis of network situations, it is desirable to have records that help to understand the conditions that let to or were caused by a certain situation.

Identified Requirements:

- Device and enabling services that are pre-programmed on CoBIs nodes and allow remote access, programming and low-level network monitoring

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Technical Requirements from a Service Oriented	Deliverable	D401

- Means for configuration of enabling and device services: Configuration data tells them to reactively and/or proactive monitor the state of node hardware, software as well as service quality and RF link quality
- Find metrics for service quality, allow service specific adaptation

4.3 Mapping Services to Nodes

After implementing an atomic service or composing a compound service and after we know the state of a service, we can start mapping the services to certain nodes. The decision, which service must run on which node is done by a special system entity: the *mapper*. This component takes the technical requirements that are specified in a service's technical requirements record and tries to find an optimal deployment pattern based on the WSNs current status.

Of course, the user should be able to specify how services should be mapped to nodes. He should be able to choose of one of the following alternatives: Deploy a service

1. on all nodes
2. on one explicitly specified node
3. on a set of explicitly specified nodes
4. on n nodes chosen by the mapper
5. on p percent of the nodes chosen by the mapper

In the cases 1-3 no automatic mapping would take place and the services would be deployed exactly as specified. In the cases 4 and 5 the mapper would decide where to deploy the service.

The mapper should also take care of updates. It should maintain a list of deployed services and detect when a new version of a service is available in the service repository. It should then alert the user or (if configured so) automatically update all instances of the service. The task of updating will be carried out by the injector component described in the next paragraph but initiated by the mapper.

Identified Requirements:

- Find an appropriate deployment pattern based on the requirements of a service and the current state of the network if the user specifies mapping of services to nodes with a certain degree of freedom
- Allow reconfiguration of the network by stopping and deleting services on a node and applying new ones

4.4 Injection and Start of a Service

When the mapping component has decided on which node(s) a service must be deployed, it forwards this decision to the injector component. The injector transfers the service's implementation to the picked node(s). It does this by accessing a *programming service* on a node

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Technical Requirements from a Service Oriented	Deliverable	D401

that receives the incoming implementation and stores it in the devices memory. This programming service could either just program one device at a time or run distributed in order to forward the program (or blocks of code) to all nodes that need to be programmed.

Programming a subset of nodes of a WSN at the same time is a special case of reliable multicast. All program data emitted by a gateway node between the backend system and the WSN must be transported reliably to the given subset of nodes.

The injector service also offers a converse action for stopping and removing services. When a service is stopped on a node, its execution is halted but it remains in the nodes memory. When it is started again, the states of the service are reset and execution begins just like right after deployment. If a service is removed from a node it is stopped and the program and state memory of the service is freed. By removing a service and deploying another one, an existing service can be updated to a new version or replaced by another one.

Once a service is deployed on a node, it remains inactive but ready to run. When the programming service receives a wake up signal for the service, it initializes and starts it.

Identified Requirements:

- Enabling or device service for injecting (programming) another service into a sensor node. Optionally, a distributed middleware service that intelligently and reliably forwards packets to the nodes that have to be programmed
- Gateway hardware and service that connects the conventional hardware to the WSN
- Programming service must also be able to start, stop and delete services on a node
- Implicit: An engine capable of executing atomic and compound services

4.5 Monitoring a Running System

Monitoring a CoBIs system during operation - after injecting and starting all necessary services - is quite similar to pre-deployment monitoring, although it serves different purposes and can deal with more abstract parameters. It keeps the user informed about actual and potential unfavourable situations.

In-operation monitoring can rely on the monitoring services included in the set of enabling or device services. These should provide basic, low-level monitoring of the system. The monitoring parameters should comprise hardware health diagnosis (self-test results etc.), statistical data of the software (e.g. messages in in/out queue of a service, names and runtime of all services on a node), as well as raw sensor readings for diagnostic purposes.

The basic monitoring services (implemented as device or enabling services) should report problematic events like link breakage, low battery power failure of self test etc. For reporting these situations we need an *alerting and logging service*. This should be considered an enabling service that should run on each node as well as in a backend entity. The instance in the backend is the target for all logging and alerting messages and it forwards the data to other backend components. Local logging on a device is not feasible because of local memory limitations.

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Technical Requirements from a Service Oriented	Deliverable	D401

If the functionality of the basic monitoring services is not sufficient for the needs of the application, additional application-specific monitoring services can be developed and deployed. Those can be tailored to the information needs of the application. The custom monitoring service can make use of the basic alerting and logging services. If basic alerting and logging lacks some functionality, this component can also be extended by designing and deploying a custom logging and alerting service.

Identified Requirements:

- Monitoring, logging and alerting services (implemented as enabling services), that forward data to a corresponding backend component.
- Possibility to develop and deploy application-specific custom logging and alerting services

4.6 Service Remapping and Auto-Deployment during Operation

The data generated by the monitoring service (basic or application-specific) can be used to optimize a CoBIs system automatically without user interaction. The basic idea is to use the mapping component introduced in section 4.3 at regular intervals or if a potentially harmful situation is detected. This way, the system can react to changes in e.g. node connectivity or battery status and relocate certain services. Figure 6 depicts the service remapping process.

To enable service remapping at operation time, the service description must contain additional data. In addition to the initial deployment rules shown in section 4.3, a service has to be described as *fixed* (not relocatable to another node), *moving* (relocatable) or *partly moving* (relocatable within a given set of nodes). If a service is specified to be deployed on all nodes or a certain percentage of nodes, it is automatically assumed moving. If all services running in a CoBIs network are specified as fixed, re-mapping will not occur.

A variant of service remapping is auto-deployment. As we specified in section 4.3 a user can tell the system that a service should be deployed on all nodes or a certain percentage of nodes. If a new node appears in the network (e.g. because a new chemical barrel has been equipped with a CoBIs node), and this node is still in its initial state only running enabling services, the mapping entity should be informed about the new node and can then deploy the service on the new node if necessary.

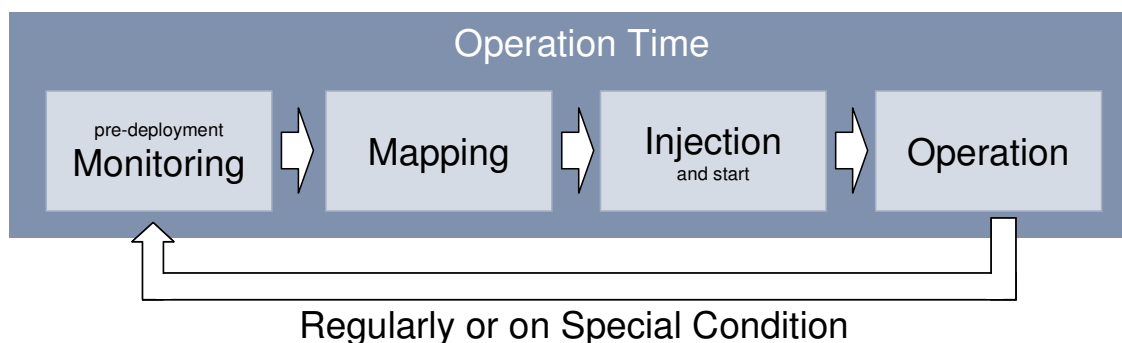


Figure 6 - Re-mapping of Services during Operation

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Technical Requirements from a Service Oriented	Deliverable	D401

Identified Requirements:

- Component that triggers pre-deployment monitoring in case of a special situation detected by the monitoring service or on a regular basis
- Method of specifying special situations that should be logged / alerted or shall trigger service re-mapping
- Regular or event-driven re-mapping of the services to nodes

4.7 Configuring a Running System

Since CoBIs systems are service enabled they can have an easy to understand configuration model. Configuring a CoBIs system means nothing but adjusting parameters of services.

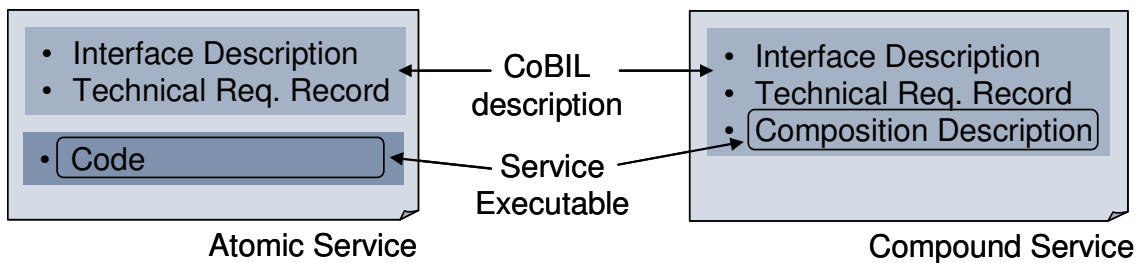


Figure 7 - A Service in the Service Repository

As we already discussed, a service in the central service repository consists three parts: First, a description in CoBIL. Second an implementation (in case of an atomic service) or model how it is composed out of other services (as a part of the CoBIL description). Third, every CoBIL description of a service contains its interface, i.e. a list of public parameters that can be read or written by other services and their corresponding access rights.

With this easy configuration model we can implement an enabling service that can change parameters of services. This leads to a straightforward, asynchronous communication model between services that run on one node.

It has to be evaluated if it is feasible to use the method of asynchronous parameter setting also for inter-node communication between services. This way, the enabling service used for remote configuration could also be reused as a remote communication service. It could use the appropriate device service for communication to forward the parameter changes.

Identified Requirements:

- Definition of service parameters and access rights (in CoBIL description)
- Basic configuration service (implemented as enabling service) on each node that

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Technical Requirements from a Service Oriented	Deliverable	D401

- receives parameter updates and applies them to a service instance
- answers to queries for server parameters
- respects the access restrictions in the services description

4.8 Updating a System

To update a service running in a CoBIs system, a user would develop a new version of an atomic or compound service. When the new version of a service is injected into the service registry, the mapper realizes the update. Based on the administrator's choice stored in the system, the mapper can now update all instances of the service on its own or inform the application to issue a user interaction and obtain a confirmation for the update.

Identified Requirements:

- Support for version identifiers in the repository
- Basic reporting service that reports the names (or ids) and versions of all services currently running on a node

4.9 Licensing and Security

Although they are out of the scope of this project, licensing and security are important issues for CoBIs networks. However, because of limited project time and because this functionality can be easily added to the final CoBIs system, we will treat the implementation of these subjects with low priority. Nevertheless, we want to discuss some aspects of the two points:

One can think of many kinds of licensing models for CoBIs networks. A licence could e.g. allow only n events at a node over a certain period of time, and you would have to upgrade your licence if your system would grow and more events take place. A licence could also be restricted to a device class, or a certain degree of customization. It could be an expiring evaluation licence or a combination of all these models.

Security support for a CoBIs network would comprise user authorization, roles with access rights, authentication and password management.

Project Title	Collaborative Business Items	Project No.	004270
Project Acronym	CoBIs	Date	20-02-2005
Chapter	Conclusion and Future Work	Deliverable	D401

5 Conclusion and Future Work

The preceding report systematically listed the management and maintenance features which are required to administrate large scale CoBIs systems in real world business processes. A first result of the CoBIs WP4 research is the structured survey of dedicated management and maintenance services, which will be the basis for the development of the CoBIs administration tools. Having the broadest install base in today's enterprises, we want the CoBIs system administration software to run on Microsoft Windows based systems.

The derived business, enabling, management, and device services can be classified according to the following nine service tasks: modelling business logic, monitoring before deploying, mapping services to nodes, service injection and calls, monitoring a running system, service remapping and auto-deployment, configuration, updating, and security features. The proposed set of services has to be validated and refined by investigating the CoBIs application trials.

In the upcoming research activities further feedback about the system implementation will be gained by looking at the experiences on the end-user side. The development of deployment techniques and monitoring tools for basic- and device-services has to be coordinated with the abstraction layer developed by WP2 and WP3. The higher-level administration services mainly have to accommodate to the application specific features examined by WP1.

At the end of the second project phase the management and maintenance concept for CoBIs environments is established, ready-to-use, and implemented in the final application trials.