EUROPEAN COMMISSION

Thematic Priority:
SIXTH FRAMEWORK PROGRAM

Priority 2.5.3
INFORMATION SOCIETY TECHNOLOGIES
Unit G3 Embedded Systems

Project Acronym:

SOCRADES

Project Full Title:

Service-Oriented Cross-layer infRAstructure for Distributed smart Embedded devices

Proposal/Contract No: EU FP6 IST-5-034116 IP SOCRADES

Deliverable D1.2

Requirements of end users and component vendors/system integrators

Status: Final

Dissemination Level: CO

Date: 20.04.2007

Organization Name of the Lead Contractor for this Deliverable: ifak

Status Description:

Scheduled completion date[1]:	31.03.2007	Actual completion date[2]:	20.04.2007
Short document description:	This document describes the basic requirements for networked embedded systems.		
Author(s) deliverable:	Thomas Bangemann (ifak) Francois Jammes (SE) Antione Mensch (SE) Martin Strand (ABB) Oliver Baecker (SAP) Stamatis Karnouskos (SAP) Luciana Moreira Sa de Souza (SAP) Rob Harrison (Lboro) Radmehr Monfared (Lboro) Axel Klostermeyer (Siemens) Mats Lillqvist (FLEX) Axel Bepperling (SE) Jerker Delsing (LTU) Harm Smit (SE) Patrik Spiess (SAP) Fabrice Depeisses (SE)	**Report/deliverable classification:** [x] Deliverable [ ] Three-Monthly Activity Report [ ] Six-Monthly Activity Report	
[ ] [ ] Partner ↓ ↓ Contributions / Peer reviews	[x] [x] Schneider Electric [x] [x] ABB [x] [ ] APS GmbH [ ] [ ] Boliden AB [x] [ ] FlexLink Automation Oy. [x] [x] Institut f. Automation und Kommunikation e.V. Magdeburg [x] [ ] Kungliga Tekniska Högskolan	[x] [x] Loughborough University [x] [ ] Luleå University of Technology [ ] [x] Politecnico di Milano [x] [x] SAP AG [x] [x] Siemens AG [ ] [x] Tampere University of Technology [x] [ ] Jaguar Cars Ltd. [ ] [ ] ARM Ltd.	
Peer review approval : [x]	Approved Rejected (improve as specified hereunder)	Date:	27.03.2007
Suggested improvements:			

[1] As defined in the DoW

[2] Scheduled date for approval

Executive Summary

This deliverable D1.2 is intended to lay down the fundamentals for the design of the SOCRADES architecture, its components and characteristics while investigating potential use cases for networked embedded components and the explanation of a first set of basic requirements visible at this early stage of the project.

Whereas use cases are described in D1.2 and its associated "Annex to D1.2" very extensively, requirements will have to be improved and extended after expert work done within the dedicated work packages. Consequently, in this document a general approach for requirements definition has been defined and applied first. The same approach will be followed by other work packages specifying additional requirements. This approach has been defined to be able to detail requirements after having had first discussions on architectural and technical subjects.

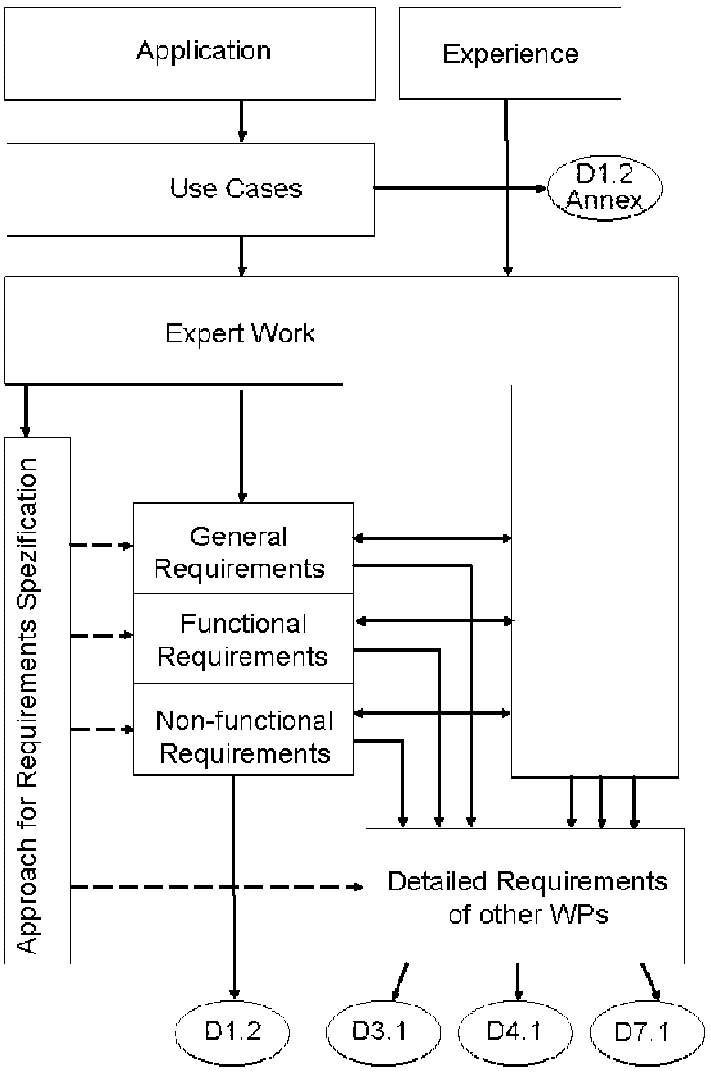

Figure 1: Overall requirements specification strategy

The strategy illustrated above is reflected by the structure of the overall document D1.2 and its annex.

Before starting to investigate use cases for networked embedded devices in the annex to D1.2 and referenced in chapter 1, a general guideline for use case definition has been defined. This is to guarantee easy reading and understanding different contributions coming from different partners representing different domains. To cope with all the life time of targeted components, different life cycle phases have been identified to be considered by the selection of use cases. This has been considered in several fields of application for networked embedded devices.

Chapter 2 is dedicated to requirements specification on distributed smart embedded devices and their use. Following the same idea of a common approach for defining all the requirements (as done for use cases), section 2.1 starts with the introduction of the approach applicable to requirements specification in D1.2, to be used as well in other work packages. Requirements will become defined in a hierarchical manner starting with general or use case related requirements. Discussion of these requirements will lead to more specific functional and non-functional requirements. Using relational attributes, it will be possible to highlight dependencies and relations between requirements themselves as defined in D1.2 and respective deliverables of other work packages as well as to use cases.

Requirements are numbered and described in sections 2.2 to 2.5.

To have an overview and an easy approach to find requirements, a requirements catalogue (chapter 3) is placed at the end of the document, associated with a section of terms used (chapter 4) and references (chapter 5).

Table of Contents:

List of Figures:

1. Use cases for application of distributed smart embedded devices

This chapter covers the guideline for describing use cases as well as a list of use cases for distributed smart embedded (SOCRADES) devices described in the annex to this deliverable 1.2. From theses use cases, requirements will be derived. Use cases are dedicated to different branches and application tasks and are related to all dedicated automation system life cycle phases like design, operation or maintenance.

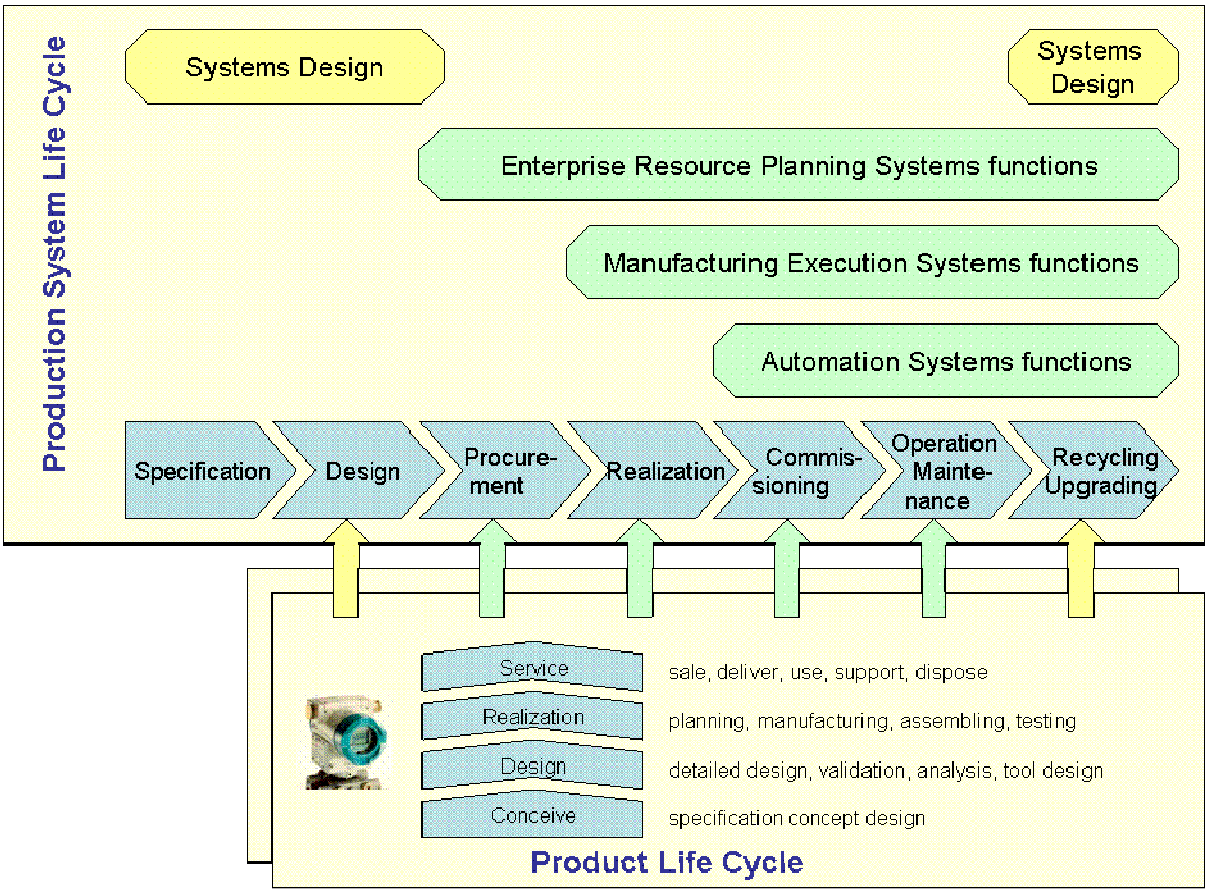

Figure 2: Automation system lifecycle

Automation life cycle phases

Phase	Phase description	Actors involved
Specification	Specify the machine from a functional point of view. Define the process and the different operating modes.	Customer, Integrator
Design	Design the system: Mechanical conception, electrical, pneumatic, software conception, network topologies, etc…	Integrator
Procurement	From the functional definitions, select and buy the technologies: Actuators, sensors, I/O devices, PLCs, etc, needed to fulfil operations.	Integrator
Realization	Build the machine, assembly of the actuators / sensors, electrical, pneumatic, and network cabling, software	Integrator

	coding and tests, integration.	
Commissioning	Installation and progressive power-up into the customer production chain.	Integrator, Customer
Operation and Maintenance	Operation and maintenance of the machine during its life.	Customer, Integrator, Maintenance Personal
Recycling / Upgrading	Upgrade or destruction of the machine. Decided by the customer.	Customer, Integrator

Actors

Actor	Description	Phases involved in
Customer	The Customer is the reference for the process to automate, and gives functional requirements in the specify phase. He will accept the machine at the end of the commissioning phase, be responsible for the operation and first level of maintenance, and decide to upgrade the machine, or recycle it at its end of life.	Specification, Commissioning, Operation and Maintenance, Recycling and Upgrading
Integrator	The designer of the machine. Involved in all the phases of the process, he is the only one who "knows" completely the machine.	All, except Operation
Operator	Customer employees who uses the machine for production.	Operation
Maintenance Personal	Customer employees who can change mechanical / electrical parts of the machine	First level of Maintenance. Higher levels are realized by the Integrator.

1.1. Guidelines for use case definition

This chapter provides the template for describing use cases. A use case defines a goal-oriented set of interactions between external actors and the system under consideration. Actors are parties outside the system that interact with the system. Actors are changing across the life cycle phases of the system considered. This can be easily shown by a UML use case diagram as illustrated below.

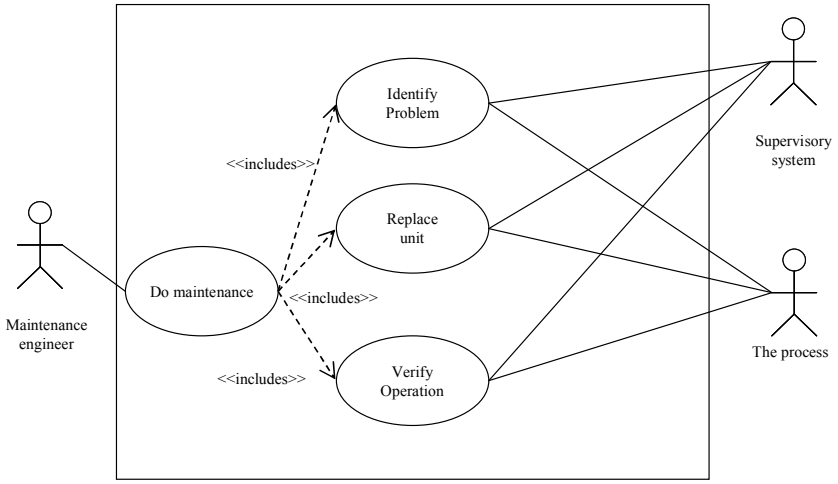

Figure 3: Example of a UML use case diagram

To be more specific, each use case will be separated into several steps.

To ease reading the use case section of this document, use cases are defined in a common manner according to the following template:

1.1.1. Use Case <UC-Id> <Use Case name>

The UC-Id is a unique identifier for the use case. The UC-Id will be build by a "UC" and the number of the chapter, followed by a consecutive number (unique for a chapter). (Example: Use Case. Industrial Automation. Use Case no 3 → UC-1.2.03).

The Use Case name is a concise description of the use case that will be used as a title.

1.1.1.1. Description

Each use case should have a description that describes the main business goals of the use case.

1.1.1.2. Step list

The step list represents the sequence of interactions necessary to successfully meet the goals of the use case. The interactions between the system and actors are structured into one or more steps which are expressed in natural language.

Each step is identified by a number, has a name and may optionally be assigned a focus.

Conditional statements <Variation> can be used to express alternate paths variations on the manner or mode in which a step may happen. The use of <Variation> is optional.

No	V.No.	Name	Focus
1			
2			
	2a		
	2b		
3			
4			
	4a		

	4b		
	4c		
General variation			
		Variation 1:	
		Variation 2:	

1.1.1.3. Step description

Step <No>: <Step Name> description

A description that describes the details of the step shall be added here.

Step <No> Variation <V.No.>: <Variation Name> description

A description that describes the details of the variation of the step shall be added here. A variation may also be applied to the overall list of steps.

1.2. Summary of use cases

This section introduces a set of use cases, which are a fundamental basis for retrieving requirements on networked embedded devices.

While having project partner coming from, and representing different branches or domains a broad range of application fields is covered by theses use cases. This is the basis for finally being able to exploit SOCRADES results in a broad range of applications.

This section and all its sub-sections provide introductions to different application domains and lists of related use cases. As this D1.2 is explicitly dedicated to requirements specification, the detailed use case descriptions have become moved to the annex, although certain effort was spent to the identification and discussion of use cases applicable to different phases of the life cycle of systems in the domains considered.

The use cases presented within the annex to D1.2 are separated into several sections, each focusing a specific field of application, as there are:

1.2.1. Use Cases from industrial automation

Here, industrial automation is represented by the automation of a filling machine used in food industry to fill yoghurt jars. The machine is composed of a conveyor belt transporting jars, and filling system with a tank.

For this example, the overall life cycle, starting from the application design and covering commissioning, operation, maintenance till redesign has been analyzed as a basis for use case description. This resulted in the following use cases:

- Use Case UC1.2.01 Application design
- Use Case UC1.2.02 Program application: coding the HMI component
- Use Case UC1.2.03 Program application: coding the Tank component
- Use Case UC1.2.04 Program application: coding the Conveyor component
- Use Case UC1.2.05 Program application: coding the analogue sensors components
- Use Case UC1.2.06 Program application: Final coding

- Use Case UC1.2.07 Build the physical device implementation
- Use Case UC1.2.08 Application deployment
- Use Case UC1.2.09 SCADA access to the application
- Use Case UC1.2.10 Commissioning with PDA
- Use Case UC1.2.11 Maintenance intervention with PDA
- Use Case UC1.2.12 Replace a faulty device
- Use Case UC1.2.13 Reconfigure the machine
- Use Case UC1.2.14 Stop part of the machine for cleaning intervention
- Use Case UC1.2.15 Redesign application for redundancy
- Use Case UC1.2.16 Agent-based Control

1.2.2. Use Cases from flexible manufacturing (car production)

The system considered for use case definition is typical of assembly automation machinery used in automotive engine manufacture, where assembly stations are link via free conveyors. The overall context for the application is described and a typical set of components and their functional requirements are outlined for a typical functional module in such a system.

After presenting the system characteristics, the annex to D1.2 identifies use cases, which are mainly dedicated to the engineering related phases of the life cycle. This resulted in the following use cases:

- Use Case UC1.3.01 Automation System Specification
- Use Case UC1.3.02 Automation System Design
- Use Case UC1.3.03 Automation System Realisation
- Use Case UC1.3.04 Automation System Commissioning
- Use Case UC1.3.05 Reconfiguration of Automation System

1.2.3. Use Cases from process control

Process industries like, paper mills, chemical plants, mining industries, oil refineries, is characterized by continuous flow operations. Historically the production parameters have been rather static and the main variable has been the volume to produce. Now days, the quality and other product characteristics are also variables that vary frequently due to the need to adapt to the market and serve customer demands. The production equipment and the control of it have become more flexible. This has been considered while identifying the following use cases, dedicated to all the life cycle phases introduced:

- Use Case UC1.4.01 Specification
- Use Case UC1.4.02 Engineering
- Use Case UC1.4.03 Installation
- Use Case UC1.4.04 Commissioning
- Use Case UC1.4.05 Operation
- Use Case UC1.4.06 Maintenance and repair

1.2.4. Use Cases regarding high level management through MES/ERP

Integration of enterprise or operational management related information, coming from smart embedded devices, is one major focus of the SOCRADES project. This will enable advanced or even new concepts of business and production management processes. The use cases listed below describe some of those concepts:

- Use Case UC1.5.01 Monitor system state
- Use Case UC1.5.02 Process system alerts
- Use Case UC1.5.03 Invoke SOCRADES services
- Use Case UC1.5.04 Update embedded software services
- Use Case UC1.5.05 Remote maintenance
- Use Case UC1.5.06 Predictive maintenance
- Use Case UC1.5.07 Avoidance of run-time deficiencies
- Use Case UC1.5.08 Feature upselling

1.2.5. Use Cases regarding wireless and nomadic applications

SOCRADES systems shall support applications based on wired, wireless and mixed networks transparent to the user. Nevertheless wireless sensor networks or wireless parts in mixed networks have certain influence on all the life cycle phases of the system, which has to be considered for the design of SOCRADES components. The following use cases have been defined as a basis for considering these specific characteristics:

- Use Case UC1.6.01 plan process environment
- Use Case UC1.6.02 plan type of application
- Use Case UC1.6.03 plan communication
- Use Case UC1.6.04 install network
- Use Case UC1.6.05 operate plant
- Use Case UC1.6.06 plant diagnosis

1.2.6. Use Cases regarding flexible manufacturing applications

The most dynamic approach regarding application and its re-configuration of the automation system, considered within D1.2, is the one of a flexible manufacturing system in the domain of electronics assembly. The annex to D1.2 provides a description of a typical PC assembly system from a factory floor perspective. Then, a description of the interactions that occur between an Enterprise Resource Planning (ERP) type of system and the Manufacturing Execution System (MES) that is in charge of supervising and coordinating the factory-floor activities is given. Finally, a detailed description of the equipment modules used to build the system is given, providing details on the operations that each provides using Web Services.

This flexible manufacturing is described within the use cases listed below:

- Use Case UC1.7.01 Conveyor transfer
- Use Case UC1.7.02 Cross unit transfer
- Use Case UC1.7.03 Lifter transfer
- Use Case UC1.7.04 Pallet at routing point
- Use Case UC1.7.05 Transfer into junction
- Use Case UC1.7.06 Pallet loading

- Use Case UC1.7.07 Pallet unloading
- Use Case UC1.7.08 Manual assembly operation
- Use Case UC1.7.09 Robot assembly operation
- Use Case UC1.7.10 Test/inspection/repair operation
- Use Case UC1.7.11 Product order
- Use Case UC1.7.12 Product order modified/deleted
- Use Case UC1.7.13 Product order complete
- Use Case UC1.7.14 Product order failed
- Use Case UC1.7.15 Order dispatch
- Use Case UC1.7.16 Initial line configuration
- Use Case UC1.7.17 Line re-configuration: new conveyor path
- Use Case UC1.7.18 Line re-configuration: station goes offline
- Use Case UC1.7.19 Line re-configuration: new station online

2. Requirements on distributed smart embedded devices and their use

Based on experience of the project partners, this chapter will analyze requirements regarding:

- unsolved issues of connecting embedded systems through heterogeneous wired and wireless networks;
- the needs for flexibly integrating networked embedded systems in real-time, collaborative business environments;
- integration of embedded components with higher level applications;
- engineering support (e.g. commissioning and diagnosis issues).

The overall requirements specification starts with requirements defined within this deliverable D1.2 and will become extended and detailed by requirements listed in D3.1, D4.1 and D7.1. This improvement process of requirements guarantees requirements of high quality.

The concomitant requirements will be derived in conjunction with those emanating from end users, component vendors and system integrators.

These requirements will be defined following a unified approach of requirements definition as defined hereunder.

In order to direct requirements to precise aspects of the overall problem addressed by the SOCRADES project, it is useful to define a general architecture identifying the main components of the SOCRADES approach.

The SOCRADES approach is to create system intelligence by a large population of small and smart networked embedded devices at a high level of granularity, as opposed to the traditional approach of focusing intelligence on a few large and monolithic applications. This increased granularity of intelligence distributed among loosely coupled intelligent physical objects facilitates the adaptability and re-configurability of the system, allowing it to meet business demands not foreseen at the time of design.

In order to achieve this vision, the proposed SOCRADES service-oriented architecture relies on three different layers, as shown in the figure below:

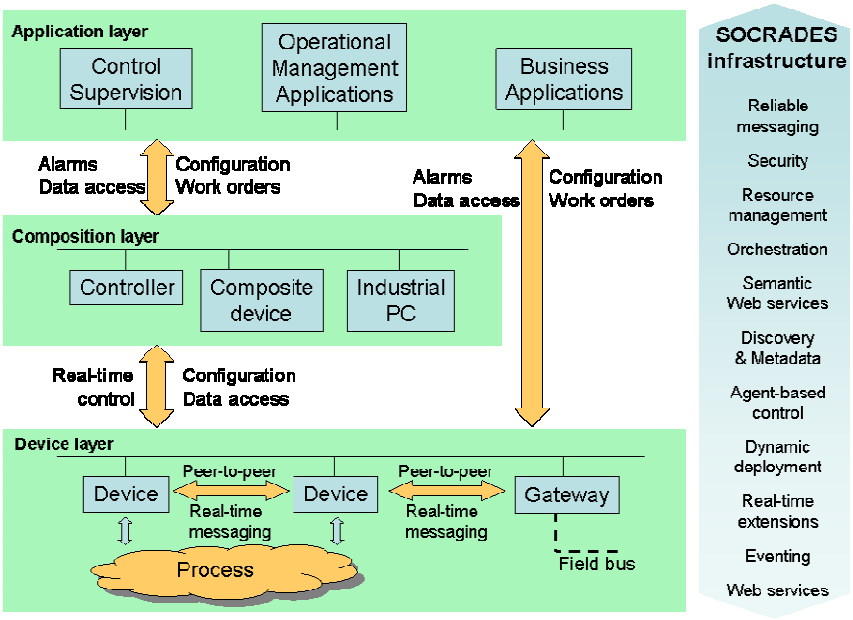

Figure 4: Overall SOCRADES infrastructure

The **device layer** contains the small, smart devices that provide most of the distributed intelligence through their exposed services. Apart from devices supporting SOCRADES functionality classical devices may be integrated using SOCRADES gateways individually or gateways connecting fieldbus networks.

The **composition layer** contains the intermediate systems that provide added value (e.g. control, monitoring, diagnosis, …) services by combining the capabilities of the smart devices (SOCRADES devices or devices accessed through gateways). Due to the distribution of functionalities towards the smart devices, these composite systems are much more flexible and agile than the monolithic systems they replace.

The **application layer** contains both applications that are traditionally strongly connected to the lower layers, such as supervision, as well as applications that do not have until now direct access to those layers, such as business applications (ERP level) or operational management (MES level) like maintenance. The exposure of device functionalities as services and the use of standard communication protocols allow such applications to seamlessly access any device on the network.

These three layers are connected together and interact through the SOCRADES infrastructure:

- **Device layer:** Smart devices directly connected to the physical system or process under control and exposing their capabilities as services belong to this layer, as do gateways encapsulating field bus devices not able to expose their capabilities as services. Key features of the SOCRADES infrastructure used by this layer include (i) ability to expose device functionalities as *Web services and events* (ii) *peer-to-peer, real-time interactions* between devices using these Web services, invocations and events, leading to new ways of implementing *distributed, agent-based control*, (iii) plug-and-play capabilities based on *discovery and dynamic metadata exchange* protocols, and (iv) *dynamic deployment facilities* providing ways of programming and reprogramming the devices to adapt them to their environment.
- **Composition layer:** Higher-level devices, such as controllers, machines (composite devices), special purpose equipment able to execute complex functions and industrial PCs belong to this layer. They rely on the composition and aggregation of the services exposed by the device layer to implement higher-level functionalities for control and management purpose. In addition to those already described, key features of the SOCRADES infrastructure used by this layer include (i) *semantic Web*

services description and discovery and (ii) *service orchestration* and aggregation languages and engines, including real-time orchestration based on IEC 61131 languages.

- **Application layer:** applications in this layer benefit from the following features provided by the SOCRADES infrastructure: (i) support for Web services-based *resource access and management*, allowing applications to access device resources directly or through SOCRADES enabling gateways and (ii) secure and reliable communication protocols compliant with enterprise policies.

The SOCRADES infrastructure acts as a "service bus" connecting devices together and with control, management and enterprise applications. It features standard network equipments and IP-based protocols, value-added protocols and generic services, in order to provide connectivity and quality of service for devices.

The infrastructure is composed of connected nodes, which can be any type of processing equipments: devices, gateways, controllers, workstations and even enterprise servers. Each node acts as a service container that can run both application services and generic system services, depending on its capabilities and the application requirements. Nodes can communicate together through various protocols, also depending on their capabilities and requirements, thus providing adaptive quality of service. The following figure shows the infrastructure-centric view of SOCRADES containers:

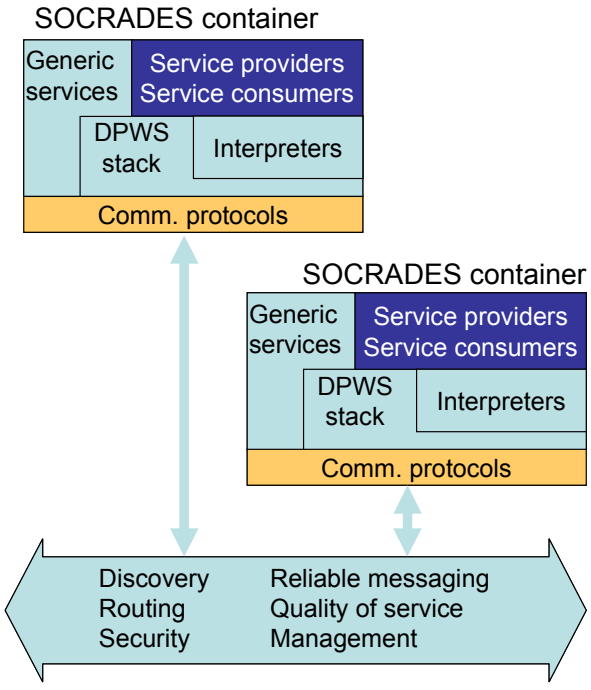

Figure 5: SOCRADES containers are enabling SOA

While the bus at the bottom is depicted in the figure as providing some advanced capabilities, such as discovery, security or management, in reality those capabilities are provided by the collaboration of individual containers in which the appropriate infrastructure elements have been deployed. For instance, secure communications can be provided between two devices only if both devices use the security extensions of the Web services (DPWS) stack, and if appropriate authentication information has been deployed beforehand, either on each device or in an external authorisation server.

In addition to the Web services stack, additional components can be included in containers:

- Interpreters used to execute dynamically loaded services, using for instance control languages such as IEC 61131 languages, or orchestration languages such as BPEL.

- Generic services, including extended discovery services, eventing services or management services.

SOCRADES containers are modular: only required components are deployed in a given container, thus allowing the footprint of the infrastructure to be adjusted to the exact needs and capabilities of the application.

2.1. Approach for requirements specification

The requirements specification approach covers the overall process of gathering, expressing, organizing, tracing, analyzing, reviewing, agreeing, changing and validating requirement statements, which will start with deliverable D1.2 and be continued by D3.1, D4.1 and D7.1. Requirements listed here are based on experience

- coming from real life applications of different branches,
- of different personnel roles (user, system integrators, development stuff, application and technology experts, …)
- different WP's of the SOCRADES project

This deliverable was organised around the assumption that it is possible to gather requirements for distributed smart embedded devices at different level. Levels can be forward mapped with the next through a chain of links that provide an useful framework for this early phase of the project. During project's progress other WP's may check requirements and their relations for easy focusing their work. During validation phase these relations will support the checking tasks.

General Requirements

General requirements are stated e.g. inside the Description of Work document. They focus on major problems the SOCRADES project is targeting, and its general motivation.

Functional requirements

Functional requirements are driven and traced from business requirements and usually provide precise behavioural statements ("system should do this").

Non-functional requirements

Non-functional requirements are driven from functional requirements, but are related to more internal behaviour (e.g. performance or quality) or constraints of the service provided (e.g. standard compliance).

Traceability is a key concept in requirements management and enables the traversal of related requirements from the general requirements through functional and non-functional requirements, and into design specification and implementation activities. Many of the concerns of requirements management exist to enable traceability. It is through this discipline that enables to check and document:

- how requirements are satisfied and
- how requirements are tested.

Following the basic requirement template structure is presented and explained. This template defines the unified structure and content of all requirement statements contained within this document or any other SOCRADES deliverable.

Requirement

<short title of the requirement>					
Description					
<detailed description of the requirement>					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(Dx.y)a.b.c.de	WPx, WPy, …	UC-1.2.03, R(Dx.y)a.b.c.df, …	Thomas Bangemann (ifak)	dd.mm.yyyy	
Remarks					
<explanatory comments of the author or remarks of other persons, description of the requirement's sources (end user, literature, discussions, …), …>					

A description of the elements of the requirements template is given below:

Requirement

A concise description of the requirement that will be used as a title. The title must contain a priority classification differentiated between:

- MUST (This word means that the definition is an absolute requirement of the specification.), RFC 2119
- SHOULD (This word, or the adjective "RECOMMENDED", means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.), RFC 2119
- CAN (This word, or the adjective "OPTIONAL", means that an item is truly optional.), RFC 2119

Description

A detailed description of the purpose and the objective of the requirement. This requirements element provides the specification of the essential subject matter of the system. Specifying a requirement clarifies the system's subject matter and in doing so, may trigger requirements that have not yet been thought of.

Req-Id

This element is a unique identifier for the requirement. The ID will be build by a "R", followed by the abbreviation of the deliverable put into brackets and the number of the chapter, followed by a consecutive number (unique for a chapter). A point will separate the hierarchical identifier as in paragraph names. In case sub-chapters are used also the ID should maintain this hierarchical structure. (Example: Requirement within D3.1, General Requirement, Requirements on Wireless Technology, Requirement no 8 → R(D3.1)2.2.1.08)

Target WP's

This element provides the information about WP's targeted by this requirement. WP's listed here must care about the resolution of this requirement.

Relations

This element lists other requirements and use cases. This requirement is related to or even dependent (e.g. that the requirements can only be fulfilled if another requirement is also fulfilled). Related requirements are identified by their Req-Id, use cases by their UC-Id.

Author

This element consists of the name of the originator of the requirements and the abbreviation of the partners name he is representing (e.g. Thomas Bangemann (ifak)).

Date

This element provides the date, the requirement has been issued.

Remarks

This element may be used to provide additional information related to this requirement. Such information may be the original source (end user, literature, discussions, …) or some unspecific comment.

2.2. General requirements

Requirement					
The SOCRADES architecture MUST support heterogeneous (wired, wireless) network environments.					
Description					
Today it is not foreseeable what the real application of an individual SOCRADES component will be. Today's applications demonstrate that the geographic and environmental conditions are of enormous diversity. Many of those applications can be realised through combination of wired and wireless sub-systems. In case this diversity is not considered during the design phase of the system architecture, extremely high effort is needed during later phases for adaptation of components. This shall be avoided by appropriate design of the SOCRADES system architecture. The SOCRADES architecture MUST consecutively support heterogeneous (wired, wireless) network environments.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.2.01	WP2, WP3, WP7	R(D1.2)2.2.04, R(D1.2)2.2.06, R(D1.2)2.3.6.x	Thomas Bangemann (ifak)	29.01.2007	
Remarks					

Requirement
Engineering of SOCRADES components SHOULD support composition of application task related services.

Description
Functionality of a component firstly depends on the hardware the component is representing. This is the natural case for sensors and actuators. Or, more generally, devices with process interface. General purpose controllers like a PLC or a modular I/O device are more flexible. Secondly functionality depends on operating systems capabilities and build-in software representing the devices physical functions and those functions depending on e.g. measurement principles. Taking those limitations under consideration, it is possible to scale and modify services provided by a component (representing a device or machine) through service intelligence built by aggregating the incremental intelligence offered by small smart devices.

Req-Id	Target WP's	Relations	Author	Date	
R(D1.2)2.2.02	WP2, WP5, WP7	R(D1.2)2.3.8.01, R(D1.2)2.3.8.02, R(D1.2)2.3.1.04, R(D1.2)2.3.1.05, R(D1.2)2.3.2.04, R(D1.2)2.3.3.01, R(D1.2)2.3.3.02	Thomas Bangemann (ifak)	29.01.2007	

Remarks

Requirement
Engineering process SHOULD be suitable to non-high-skilled users

Description
Today engineering of an automation system is a several steps approach (hardware configuration, communication engineering, application engineering). This situation should be improved after an introductory phase of the SOCRADES system architecture. The use of tools SHOULD be adapted or adaptable to the abilities of people who do the appropriate engineering task today. As the introduction and improvement of each new technology needs several time this requirement targets a time frame of some years after termination of the project.

Req-Id	Target WP's	Relations	Author	Date	
R(D1.2)2.2.03	WP7	R(D1.2)2.3.8.x	Thomas Bangemann (ifak)	29.01.2007	

Remarks

Requirement
Functional behaviour SHOULD be independent on whether cabling or wireless technologies are used for networking of embedded devices.

Description
Replacement of a wired network segment by a wireless network should not influence the overall functional

behaviour. The project SHOULD consider degradation regarding QoS and provide basic roles for engineering.

Quality of Services is even more essential for application design and finally control applications.

Req-Id	Target WP's	Relations	Author	Date	
R(D1.2)2.2.04	WP2, WP3, WP4	R(D1.2)2.2.04, R(D1.2)2.2.06, R(D1.2)2.3.6.x	Thomas Bangemann (ifak)	29.01.2007	

Remarks

Requirement

Interoperability MUST be supported between heterogeneous devices deployed in various platforms and networking technologies

Description

Today's applications demonstrate that the geographic and environmental conditions and dimensions are of enormous diversity. Applications can be small manufacturing cells of a few square meters, chemical plants of square kilometres or even gas or electricity distribution applications ranging over several 10 or 100 kilometres.

This requires applications spanning different physical network types. From the application point of view this fact is irrelevant as long as QoS are appropriate.

In case this diversity is not considered during the design phase of the system architecture, extremely high effort is needed during later phases for adaptation of components. This shall be avoided by appropriate design of the SOCRADES system architecture. The SOCRADES architecture MUST consecutively support heterogeneous network environments and may be build on top of different protocol layers.

Req-Id	Target WP's	Relations	Author	Date	
R(D1.2)2.2.05	WP2, WP7	R(D1.2)2.3.8.07	Thomas Bangemann (ifak)	29.01.2007	

Remarks

Requirement

Basic concepts and roles MUST be developed within the SOCRADES project for control over wireless links.

Description

The SOCRADES solution shall be suitable to a wide range of applications and industries ranging from discrete manufacturing to process industry. A majority of the applications in process industry are control of continuous systems described by dynamic models. Those applications are implemented by means of classical automatic control today.

The successful introduction of new approaches, SoA and wireless technologies, depend on several reasons: reduction of engineering and infrastructure costs, new useful functionalities and convenient handling to name a few. The over all driving force is, however, the constant need for increased productivity in this

industry. This is manifested by constant modifications and upgrades of the automation equipment. One of the major concerns for the SOCRADES introduction in process control is to provide robust wireless control which meets the same level of control performance as of today's wired systems.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.2.06	WP3, WP4	R(D1.2)2.3.6.x	Thomas Bangemann (ifak), Martin Strand (ABB)	29.01.2007 16.03.2007	
Remarks					

Requirement					
The basic concepts of migrating from data centric access to field level devices to service oriented approach within field level SHOULD be described.					
Description					
Introduction of a new technology is easier within a new installation than in a retrofit project. But new installations may be the exception depending on the industrial branch. Most projects are retrofit installation. Retrofit means substitution of a part of an existing installation To target this large part of the installation market the SOCRADES architecture must consider integration of existing system components. This requires the support of service interfaces attached to standard field devices as well as service gateway/proxy for access to classical field devices through different communication interfaces.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.2.07	WP2, WP4, WP5, WP6	R(D1.2)2.3.8.07, R(D1.2)2.3.1.04, R(D1.2)2.4.4.01	Thomas Bangemann (ifak)	29.01.2007	
Remarks					

Requirement					
SOCRADES components MUST provide capabilities for identifying and characterising themselves to each other and to higher level systems.					
Description					
One major problem of operational service within a plant or even approaches of auto-configuration is the capability of online identification features of the installed equipment. This is not necessarily state of the art today. Identification is needed within different life cycle phases for each equipment and must be a feature for SOCRADES components.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.2.08	WP2, WP5,	R(D1.2)2.3.2.01,	Thomas	29.01.2007	

	WP6, WP7	R(D1.2)2.3.2.03, R(D1.2)2.3.5.03	Bangemann (ifak)		
Remarks					

2.3. Functional requirements

2.3.1. Functional requirements on distributed control and service orientation

We must be able to distribute control functionalities in small automation devices, thus embedding ever more intelligence and higher level functionalities into field devices. In case of simple applications, such devices could recognise themselves automatically and have peer to peer exchanges.

Requirement					
The system MUST allow high level functionalities (e.g. control) to be distributed (i.e. embedded, deployed) into devices					
Description					
We must be able to provide the capability to distribute high level functionalities (control functionality being one of them) into automation devices … (cont'ed on the following req.) This has an impact both on footprint requirements (as small devices are targeted) and performance requirements for the message exchanges, as control flow that was centralised must now be distributed.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.1.01	WP2, WP4, WP5	R(D1.2)2.3.1.02, R(D1.2)2.3.1.06, R(D1.2)2.3.6.02, R(D1.2)2.4.3.01, R(D1.2)2.4.2.02	Fabrice Depeisses (SE)	04.01.2006	
Remarks					

Requirement					
The system MUST allow peer to peer communication between devices.					
Description					
The system must have the capability of peer to peer asynchronous devices communication (non-hierarchical, flat application). Such devices could recognise themselves automatically and have peer to peer exchanges to build simple applications. These new capabilities will allow system integrators and/or machine vendors to choose which architecture best suit its requirements and constraints.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.1.02	WP2, WP3, WP4, WP5	R(D1.2)2.3.1.05, R(D1.2)2.3.6.15, R(D1.2)2.4.2.01,	Fabrice Depeisses (SE)	04.01.2006	

		R(D1.2)2.4.2.02			
Remarks					

Requirement					
The service-centric infrastructure MUST enable devices to expose their functionalities as Web services.					
Description					
Enabling services on devices provides the functional abstraction view, and high level service accesses. Web services play a critical role in plug-and-play mechanisms, as they can be described using machine-readable metadata.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.1.03	WP2, WP5, WP6, WP7	R(D1.2)2.3.1.04, R(D1.2)2.3.1.05, R(D1.2)2.3.2.x, R(D1.2)2.3.3.x, R(D1.2)2.3.5.02, R(D1.2)2.3.5.03, R(D1.2)2.3.7.01	Fabrice Depeisses (SE)	04.01.2006	
Remarks					

Requirement					
The system MUST provide the ability to dynamically assemble services to provide higher level functional capabilities.					
Description					
"Assembly" is one of the main SOA properties. The assembly (or composition) of services could become a service, which can be accessed and reused.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.1.04	WP2, WP5, WP6, WP7	R(D1.2)2.3.1.03, R(D1.2)2.3.1.05, R(D1.2)2.3.1.07, R(D1.2)2.3.3.x, R(D1.2)2.3.4.03, R(D1.2)2.3.5.02	Fabrice Depeisses (SE)	04.01.2006	
Remarks					

Requirement					
The service-centric infrastructure MUST allow service assembly to be embedded into devices.					
Description					
Service composition should not be limited to high-end systems, but should also be supported in devices.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.1.05	WP2, WP3, WP5	R(D1.2)2.4.1.01	Fabrice Depeisses (SE)	04.01.2006	
Remarks					

Requirement					
The service-centric infrastructure MUST support a publish-subscribe interaction mechanism.					
Description					
Notification mechanisms allow a component (event producer) to dynamically interact with previously unknown components (event subscribers and consumers), hence providing increased flexibility.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.1.06	WP2, WP5	R(D1.2)2.3.4.02	Fabrice Depeisses (SE)	04.01.2006	
Remarks					

Requirement					
The service-centric infrastructure MUST support a procedure-driven interaction mechanism.					
Description					
A central service should be able to invoke sequentially a set of remote services.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2.3.1.07	WP2, WP5	R(D1.2)2.3.1.04, R(D1.2)2.3.1.05	Fabrice Depeisses (SE)	14.02.2007	
Remarks					

2.3.2. Functional requirements on Plug and Play

The system must be able to address applications where machines (e.g. dose maker, water treatment) may be added / removed / stopped / reconfigured at any time to increase throughput, performance, to add new functionalities or redundant capabilities. These machines are not known in advance, must be automatically

discovered, and the application must take automatically into account their capabilities as much as possible ... and if possible without even stopping the rest of the application.

Requirement					
The system MUST support automatic discovery of newly plugged-in and previously unknown device.					
Description					
This requirement means that before interacting with new devices, an application must at least be made aware of their presence.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.2.01	WP2, WP5, WP7	R(D1.2)2.3.2.x, R(D1.2)2.3.3.02, R(D1.2)2.3.4.01, R(D1.2)2.3.5.03	Fabrice Depeisses (SE)	04.01.2006	
Remarks					

Requirement					
The system MUST support the dynamic interaction with a newly plugged-in and previously unknown device.					
Description					
A machine (composite of devices) that is not known in advance, must be automatically and dynamically discovered, and the application must take automatically into account their capabilities as much as possible, and if possible without even stopping the rest of the application. In the context, "automatically" means without human intervention; "dynamic" means that the device is able to find a new device when required rather than in advance with a static pre-configured link.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.2.02	WP2, WP5, WP7	R(D1.2)2.3.2.x, R(D1.2)2.3.3.02, R(D1.2)2.3.4.01, R(D1.2)2.3.5.03	Fabrice Depeisses (SE)	04.01.2006	
Remarks					

Requirement
The system SHOULD support semantic description capabilities, based on shared ontologies, allowing a service consumer to dynamically adapt to a service provider providing the requested functionality through an unexpected service interface.
Description
The promises of semantic Web Services are that consumers and providers can be made interoperable even when their communications interfaces differ, as long as they have the same semantic meaning. It could

provide much greater flexibility in domains where standardisation is difficult.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.2.03	WP2, WP5	R(D1.2)2.3.2.x, R(D1.2)2.3.3.02	Fabrice Depeisses (SE)	04.01.2006	
Remarks					

Requirement					
The service-centric infrastructure MUST enable the automatic binding to a service during runtime.					
Description					
A device requiring a predefined known service during runtime, has the capability to search for several service providers and then automatically and dynamically bind to the desired one.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.2.04	WP2, WP5	R(D1.2)2.3.2.x, R(D1.2)2.3.3.02	Fabrice Depeisses (SE)	14.02.2007	
Remarks					

2.3.3. Functional requirements on dynamic deployment ((re)configuration and (re)programming)

Requirement					
The service-centric infrastructure MUST support the deployment (and un-deployment) of services (and assembly of services) on devices.					
Description					
Functionalities (services) could be control functionalities, diagnostics functionalities…and they can be deploy dynamically (at any time, when it is required) on devices. The deployment must not impact the device firmware. The deployment should be done without stopping the device's operation. And of course without device reboot. It makes sense to be able to un-deploy a service once stopped (or inactive).					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.3.01	WP2, WP5,	R(D1.2)2.3.1.04,	Fabrice Depeisses	04.01.2006	

	WP7	R(D1.2)2.3.3.02	(SE)		
Remarks					

Requirement					
The service-centric infrastructure MUST support the reconfiguration of the service binding during runtime.					
Description					
The service-centric infrastructure MUST support the reconfiguration of the service binding during runtime.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.3.02	WP2, WP5	R(D1.2)2.3.3.01, R(D1.2)2.3.2.03, R(D1.2)2.3.3.04	Fabrice Depeisses (SE)	14.02.2007	
Remarks					

2.3.4. Functional requirements on management

Requirement					
The service-centric infrastructure MUST enable the notification of newly deployed and un-deployed services.					
Description					
Peer devices have to be informed of any deployed and/or un-deployed services.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.4.01	WP2, WP5, WP7	R(D1.2)2.3.1.02, R(D1.2)2.3.2.04, R(D1.2)2.3.3.02	Fabrice Depeisses (SE)	04.01.2006	
Remarks					

Requirement					
The service-centric infrastructure MUST provide the notification of service and/or device downtime or failure.					
Description					
This means that a presence protocol is required.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.4.02	WP2, WP5	R(D1.2)2.3.3.02	Fabrice Depeisses (SE)	04.01.2006	

Remarks

Requirement
The service-centric infrastructure MUST support the management of deployed (hosted) services on device.
Description
Manage a service means: • Get its status (Operational state, Faulty state…) • Start, Stop • Set/Get properties • …

Req-Id	Target WP's	Relations	Author	Date	
R(D1.2)2.3.4.03	WP2, WP5, WP7	R(D1.2)2.3.8.01	Fabrice Depeisses (SE)	04.01.2006	

Remarks

Requirement
The service-centric infrastructure MUST support the management of devices.
Description
Manage a device means: • Get its status (Operational state, Faulty state…) • Start, Stop • …

Req-Id	Target WP's	Relations	Author	Date	
R(D1.2)2.3.4.04	WP2, WP5, WP7	R(D1.2)2.3.8.01	Fabrice Depeisses (SE)	04.01.2006	

Remarks

2.3.5. Functional requirements on enterprise integration

The service-centric infrastructure must support seamless integration between devices and applications from IT.

Requirement
Devices and IT applications SHOULD be able to interact together without intermediaries and no protocols translation.

Description
A SOCRADES enabled device must allow "direct" communication from the physical sensor/actuator level throughout the machine control level up to higher levels of the business process management system (ERP/MES/SCADA). From a logical view, applications from the IT and devices "from the field" are at the same level, an ERP application can communicate directly with the devices using and sharing the same universal interoperable communication infrastructure. Alternatively legacy on non-SOCRADES enabled devices may become integrated through gateways.

Req-Id	Target WP's	Relations	Author	Date	
R(D1.2)2.3.5.01	WP2, WP5, WP6	R(D1.2)2.3.8.04, R(D1.2)2.3.1.03, R(D1.2)2.3.5.02, R(D1.2)2.3.7.01, R(D1.2)2.4.4.01	Fabrice Depeisses (SE)	04.01.2006	

Remarks

Requirement
Devices and IT applications SHOULD implement the same Web services messaging, description, and security protocols (compatible and interoperable at least).

Description
i.e. messaging protocols are SOAP and WS-Addressing, description protocols are WSDL, and WS-Policy, security protocols are WS-Security*. Since the same protocol is used to communicate from the device level to the application level, and everybody communicates at the same level, it is now extremely easy for an ERP by example to access a counter of finished products on a production chain, or for a SCADA system to access a diagnostic service, and on the user demand, to read a more precise parameter in a sensor.

Req-Id	Target WP's	Relations	Author	Date	
R(D1.2)2.3.5.02	WP2, WP5, WP6	R(D1.2)2.3.5.01, R(D1.2)2.3.8.01	Fabrice Depeisses (SE)	04.01.2006	

Remarks

Requirement
Devices and IT applications MAY implement the same Web services discovery, eventing, management and reliable protocols.

Description

Req-Id	Target WP's	Relations	Author	Date	
R(D1.2)2.3.5.03	WP2, WP5, WP6	R(D1.2)2.3.5.01, R(D1.2)2.3.8.01	Fabrice Depeisses (SE)	04.01.2006	

Remarks

2.3.6. Functional requirements on Wireless Sensor-/Actuator Networks as Special Instance of Distributed Smart Embedded Devices

For future automation solutions it might be imaginable to equip sensors and actuators with a freely programmable processor each. In many special wired / wireless networked HW/SW systems, embedded in distributed smart physical objects, so called "sensor-/actuator networks", all processors use the same communication medium and are able to communicate directly with each other according to the peer-to-peer-principle – without any coordinating instance. All (or at least the most important of) these processors then could be equipped with corresponding software based on a service oriented architecture concept as e.g. DPWS. This would enable such sensor/actuator networks to provide necessary functionalities for the wished degree of interoperability with the ERP/MES level.

Almost inescapably in context with these sensor-/actuator networks wireless technologies have to be mentioned. Though these from a theoretical point of view are not imperative for sensor-/actuator networks, in practical considerations wired sensor-/actuator networks are not of any relevance as the wiring of a high amount of sensor and actuator nodes would be difficult and not economic.

The schematic tabular evaluation of corresponding requirements can be found in the following section:

Requirement					
The System SHOULD be Low Priced					
Description					
To enable an economic application, especially sensor nodes have to be relatively low priced.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.6.1	WP 3	R(D1.2)2.3.6.6.3	A. Klostermeyer	29.01.2007	
Remarks					

Requirement					
To fulfil the requirements of process automation the system MUST be Multi Functional, for the requirements of factory automation it CAN be.					
Description					
Sensor nodes measure data, process these and transmit them to other nodes; each node simultaneously has to act as router for multi-hop connections.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	

R(D1.2)2.3.6.2	WP 3	R(D1.2)2.3.6.3	A. Klostermeyer	29.01.2007	
Remarks					

Requirement					
The concept/architecture MUST be applicable on devices with Small Resources					
Description					
Only a small amount of computing power and memory are available on typical sensors.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.6.3	WP 3	R(D1.2)2.3.6.1	A. Klostermeyer	29.01.2007	
Remarks					

Requirement					
The system SHOULD be able to work with a maximum degree of Autonomy					
Description					
The coordination of sensor nodes happens without intervention from the outside. Objective is to fulfil the respective task and to stay functional over a maximum amount of time.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.6.4	WP3	R(D1.2)2.3.6.7 - R(D1.2)2.3.6.14	A. Klostermeyer	29.01.2007	
Remarks					

Requirement					
The system SHOULD be able to fulfil Attribute Based Addressing functions					
Description					
"Where is the temperature higher than 20°C?", NOT: „Which temperature do we have at node xyz?"					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.6.5	WP 3	(R(D1.2)2.3.6.6)	A. Klostermeyer	29.01.2007	
Remarks					

Requirement
The system SHOULD fulfil the necessary functions for Location Awareness

Description					
That is, devices will know where they are, know what objects and places are nearby, and be able to communicate with other devices and servers over new, standardised protocols, such that *location* becomes a new data type in our applications - a prerequisite for attribute based addressing.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.6.6	WP3	(R(D1.2)2.3.6.5)	A. Klostermeyer	29.01.2007	
Remarks					

As wireless sensor/actuator networks in their final expansion stage may consist of several thousands of nodes and, correspondingly, should be working as autonomously as possible, so called "Self X" features will become relevant. By "Self X" a sequence of attributes is categorised which is required by modern, flexible and re-configurable production plants in general and by their corresponding networks and network components in particular.

Requirement					
The system SHOULD fulfil all/several functions of Self X					
Description					
In order to support flexible and re-configurable networks for flexible and re-configurable manufacturing systems, wireless devices should provide the capacity to self-manage a certain number of attributes.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.6.7	WP3	R(D1.2)2.3.6.7 - R(D1.2)2.3.6.14; R(D1.2)2.3.6.4	A. Klostermeyer	29.07.2007	
Remarks					

The most important characteristics will be described in the following section:

Requirement					
The system SHOULD fulfil all/several functions of Self-Organisation/ -Configuration					
Description					
Self-organising networks are able to form a network from single nodes without external support. In this context the involved participants by means of an interactive negotiation process define their respective roles and establish suitable connections and routing paths.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.6.8	WP3	R(D1.2)2.3.6.7 - R(D1.2)2.3.6.14; R(D1.2)2.3.6.4	A. Klostermeyer	29.07.2007	

Remarks

Requirement					
The system SHOULD fulfil all/several functions of Self-Stabilisation/-Healing					
Description					
A network is said to be self-healing when it is able to detect errors in single nodes or within the communication and when it is able to heal these errors by its own means (within reasonable limits).					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.6.9	WP 3	R(D1.2)2.3.6.7 - R(D1.2)2.3.6.14; R(D1.2)2.3.6.4	A. Klostermeyer	29.01.2007	
Remarks					

Requirement					
The system SHOULD fulfil all/several functions of Self-Optimisation					
Description					
Self-optimisation means the possibility of the system and of its single components to continuously search for possibilities to increase its own efficiency and performance.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.6.10	WP3	R(D1.2)2.3.6.7 - R(D1.2)2.3.6.14; R(D1.2)2.3.6.4	A. Klostermeyer	29.01.2007	
Remarks					

Requirement					
The system SHOULD fulfil all/several functions of Self-Protection					
Description					
The ability to detect (and even avoid) unauthorised access to system functions and information is called "self-protection". Corresponding systems are enabled to protect themselves against disturbances or attacks and, thus, to keep the installation running.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.6.11	WP3	R(D1.2)2.3.6.7 - R(D1.2)2.3.6.14; R(D1.2)2.3.6.4	A. Klostermeyer	29.01.2007	

Remarks

Requirement					
The system SHOULD be Self-Describing					
Description					
Self-describing systems have the ability to forward their specification autonomously to others. They posses knowledge about their feasibilities, name, location etc.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.6.12	WP3	R(D1.2)2.3.6.7 - R(D1.2)2.3.6.14; R(D1.2)2.3.6.4	A. Klostermeyer	29.01.2007	
Remarks					

Requirement					
The system SHOULD fulfil all/several functions of Self-Monitoring					
Description					
Self-Monitoring characteristics serve the purpose of failure detection and are a basic feature of self-healing systems.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.6.13	WP3	R(D1.2)2.3.6.7 - R(D1.2)2.3.6.14; R(D1.2)2.3.6.4	A. Klostermeyer	29.01.2007	
Remarks					

Requirement					
The system SHOULD fulfil all/several functions of Self-Management					
Description					
The four characteristics *self-configuration*, *-healing*, *-optimisation* and *-protection* are summarised by the term *self-management*.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.6.14	WP3	R(D1.2)2.3.6.7 - R(D1.2)2.3.6.14; R(D1.2)2.3.6.4	A. Klostermeyer	29.01.2007	

Remarks

Requirement					
It MUST be possible to bridge information between wireless and wired network domains					
Description					
The system will consist of a mix of wired and wireless network domains and it must be possible to transparently interconnect these domains. Wired domains include legacy systems.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.6.15	WP3, WP4		Martin Strand (ABB)	1.2007	
Remarks					

2.3.7. Functional requirements on generic services built into smart devices

Requirement					
The service-centric infrastructure MUST provide a diagnostics Web service for automation devices.					
Description					
This diagnostic service must be a standardised Web Service; At least, all devices should share the same diagnostic interface (e.g. standardising the device status and an event sent when a fault occurs). This web service provides a consolidated relevant view, i.e. a field device level view, a machine level view, an application level view… This Diagnostic web service properties are: • Extensible (Optional interfaces could be added as required) • Autonomous and independent of other web services					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.7.01	WP2, WP5, WP6	R(D1.2)2.3.8.03, R(D1.2)2.3.4.04, R(D1.2)2.3.6.13, R(D1.2)2.3.7.02	Fabrice Depeisses (SE)	14.02.2007	
Remarks					

Requirement
The service-centric infrastructure MUST provide a way for troubleshooting/debugging distributed application.

Description					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.7.02	WP2, WP5, WP7	R(D1.2)2.3.8.03, R(D1.2)2.3.4.04, R(D1.2)2.3.6.13, R(D1.2)2.3.7.01	Fabrice Depeisses (SE)	14.02.2007	
Remarks					

2.3.8. Functional requirements on engineering

Requirement					
The SOCRADES approach MUST support the ability to configure and reconfigure machines built from smart modules.					
Description					
In order to maximize manufacturing agility at minimum time and cost, it is vitally important to be able to reconfigure production machinery easily and quickly. It is also important that smart modules should be easily integrated with higher level enterprise systems. During the lifecycle of those modules, they will be created and supported by all the supply chain partners.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.8.01	WP7, WP3, WP4	UC1.3.x – D1.2	R. Harrison R. Monfared	12.2.2007	
Remarks					

Requirement					
Reuse of machine components MUST be maximized					
Description					
There is the requirement to compress the time for machine build related activities wherever possible through concurrency and the reuse of previous designs and also physical components. An effective virtual engineering environment is required to support such reuse activities effectively.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.8.02	WP7, WP8	UC1.3.x – D1.2 UC1 – D7.1	R. Harrison R. Monfared	12.2.2007	
Remarks					

Requirement					
Remote assistance CAN be provided by experts to rectify machine problems					
Description					
Engineering systems may inherently offer support for the rectification of machine problems from remote locations, e.g., from the machine builder and control vendor sites. Consideration of safety and security are of key importance.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.8.03	WP5, WP6, WP7	UC1 – D7.1 R(D7.1)1.16 UC2.6 –D6.1	R. Harrison R. Monfared	12.2.2007	
Remarks					

Requirement					
SOCRADES system MUST be a vendor neutral systems/open systems					
Description					
Open, vendor neutral systems are advantageous particularly to the end-user and machine builders and could have significant impact through all machine lifecycle phases. They have the potential to significantly reduce costs, training requirements and interfacing problems.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.8.04	WP3, WP4, WP7	UC1.3.x - D1.2 R(D7.1)1.14	R. Harrison R. Monfared	12.2.2007	
Remarks					

Requirement					
SOCRADES approach SHOULD provide ability for virtual engineering					
Description					
In order to reduce costs and minimize time to market a virtual engineering environment is required capable of support all aspects of the machine lifecycle. This virtual environment must be capable or accurately predicting machine behaviour.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.8.05	WP7	R(D7.1)3.2	R. Harrison R. Monfared	12.2.2007	
Remarks					

Requirement

System design MUST ensure design simplicity where possible					
Description					
Particularly from an end-user perspective, current automation system design methods are of often complex and too general purpose. Design simplicity is therefore and important goal.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.8.06	WP7, WP8	R(D7.1)1.12	R. Harrison R. Monfared	12.2.2007	
Remarks					

Requirement					
Engineering SHOULD provide effective and seamless IT systems integration					
Description					
The provision of a consistent approach to the integration of control systems with IT systems is a key retirement for cost effective enterprise wide systems, impacting across the supply chain.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.3.8.07	WP4, WP5, WP7	R(D7.1)1.12	R. Harrison R. Monfared	12.2.2007	
Remarks					

2.4. Non-functional requirements

2.4.1. Platform related non-functional requirements

Requirement					
A minimal version of the service-centric infrastructure MUST be supported by physical devices with limited hardware resources, typically a 32 bits microprocessor, with 512Kbytes of Flash memory and 96Kbytes of RAM memory.					
Description					
Including the OS, the TCP/IP stack, the container and components, a simple device can run on top of a limited-resources hardware. The typical hardware resources of the device are a 32 bits microprocessor, with 512Kbytes of Flash memory and 96Kbytes of RAM memory. The minimal version of the service-centric infrastructure MUST support at least built-in capabilities such as service & device declaration, discovery, service invocation, eventing and deployment. Obviously, the service-centric infrastructure can run on a more powerful device (such as PLC) if capabilities are expected (security, or hot swapping capabilities for the field devices).					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.4.1.01	WP5		Fabrice Depeisses (SE)	04.01.2006	

Remarks

2.4.2. Performance related non-functional requirements

Requirement					
The service-centric infrastructure MUST support high performances messaging.					
Description					
Most industrial systems require strict real time constrains. In order to accomplish it, there is a need to improve the communications between devices. Using high performances messaging (e.g. Binary XML) will reduce both the size of messages and the time required to parse them.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.4.2.01	WP2, WP5		Fabrice Depeisses (SE)	04.01.2006	
Remarks					

Requirement					
For real-time control, the system MUST support a wireless real-time network domain where the quality of control is guaranteed.					
Description					
To support process control the system must provide a network domain where the quality of control can be guaranteed even under poor SNIR and high probability of communication outage.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.4.2.02	WP4	R(D1.2)2.2.06	Jerker Delsing (LTU), Martin Strand (ABB)	01.03.2006, 16.03.2007	
Remarks					

2.4.3. Engineering related non-functional requirements

Requirement
Engineering SHOULD provide high level process description
Description
Process description needs to be done in a manner that the process engineers can directly relate to. A high-

level graphical description method is likely to be applicable.					
Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.4.3.01	WP2, WP6, WP7	R(D7.2)	R. Harrison R. Monfared	12.2.2007	
Remarks					

Requirement
High level engineering tools SHOULD support plant layout support
Description
It is desirable for engineering tools to be capable of supporting the definition of the end-user site layout to allow specification and optimization of the system.

Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.4.3.02	WP7	UC(D7.1)1.3 D7.2	R. Harrison R. Monfared	12.2.2007	
Remarks					

2.4.4. Management related non-functional requirements

Requirement
A system SHOULD provide provision for integrated production monitoring
Description
From an enterprise perspective, monitoring production rate is a key retirement to ensure that it is ramping up satisfactorily. Individual machine monitoring is also important to ensure that system reliability targets are being met. The provision of such capabilities in a seamless manner is important preferably using a common implementation approach/technology for control and IT related systems.

Req-Id	**Target WP's**	**Relations**	**Author**	**Date**	
R(D1.2)2.4.4.01	WP7, WP6	R(D7.1)	R. Harrison R. Monfared	12.2.2007	
Remarks					

3. Requirements catalogue

Req-Id	Requirement	Target WP's
R(D1.2)2.2.01	The SOCRADES architecture MUST support heterogeneous (wired, wireless) network environments.	WP2, WP3, WP7
R(D1.2)2.2.02	Engineering of SOCRADES components SHOULD support composition of application task related services.	WP2, WP5, WP7
R(D1.2)2.2.03	Engineering process SHOULD be suitable to non-high-skilled users	WP7
R(D1.2)2.2.04	Functional behaviour SHOULD be independent on whether cabling or wireless technologies are used for networking of embedded devices.	WP2, WP3, WP4
R(D1.2)2.2.05	Interoperability MUST be supported between heterogeneous devices deployed in various platforms and networking technologies	WP2, WP7
R(D1.2)2.2.06	Basic concepts and roles MUST be developed within the SOCRADES project for control over wireless links.	WP3, WP4
R(D1.2)2.2.07	The basic concepts of migrating from data centric access to field level devices to service oriented approach within field level SHOULD be described.	WP2, WP4, WP5, WP6
R(D1.2)2.2.08	SOCRADES components MUST provide capabilities for identifying and characterising themselves to each other and to higher level systems.	WP2, WP5, WP6, WP7
R(D1.2)2.3.1.01	The system MUST allow high level functionalities (e.g. control) to be distributed (i.e. embedded, deployed) into devices	WP2, WP4, WP5
R(D1.2)2.3.1.02	The system MUST allow peer to peer communication between devices.	WP2, WP3, WP4, WP5
R(D1.2)2.3.1.03	The service-centric infrastructure MUST enable devices to expose their functionalities as Web services.	WP2, WP5, WP6, WP7
R(D1.2)2.3.1.04	The system MUST provide the ability to dynamically assemble services to provide higher level functional capabilities.	WP2, WP5, WP6, WP7
R(D1.2)2.3.1.05	The service-centric infrastructure MUST allow service assembly to be embedded into devices.	WP2, WP3, WP5
R(D1.2)2.3.1.06	The service-centric infrastructure MUST support a publish-subscribe interaction mechanism.	WP2, WP5
R(D1.2)2.3.1.07	The service-centric infrastructure MUST support a procedure-driven interaction mechanism.	WP2, WP5
R(D1.2)2.3.2.01	The system MUST support automatic discovery of newly plugged-in and previously unknown device.	WP2, WP5, WP7

R(D1.2)2.3.2.02	The system MUST support the dynamic interaction with a newly plugged-in and previously unknown device.	WP2, WP5, WP7
R(D1.2)2.3.2.03	The system SHOULD support semantic description capabilities, based on shared ontologies, allowing a service consumer to dynamically adapt to a service provider providing the requested functionality through an unexpected service interface.	WP2, WP5
R(D1.2)2.3.2.04	The service-centric infrastructure MUST enable the automatic binding to a service during runtime.	WP2, WP5
R(D1.2)2.3.3.01	The service-centric infrastructure MUST support the deployment (and un-deployment) of services (and assembly of services) on devices.	WP2, WP5, WP7
R(D1.2)2.3.3.02	The service-centric infrastructure MUST support the reconfiguration of the service binding during runtime.	WP2, WP5
R(D1.2)2.3.4.01	The service-centric infrastructure MUST enable the notification of newly deployed and un-deployed services.	WP2, WP5, WP7
R(D1.2)2.3.4.02	The service-centric infrastructure MUST provide the notification of service and/or device downtime or failure.	WP2, WP5
R(D1.2)2.3.4.03	The service-centric infrastructure MUST support the management of deployed (hosted) services on device.	WP2, WP5, WP7
R(D1.2)2.3.4.04	The service-centric infrastructure MUST support the management of devices.	WP2, WP5, WP7
R(D1.2)2.3.5.01	Devices and IT applications SHOULD be able to interact together without intermediaries and no protocols translation.	WP2, WP5, WP6
R(D1.2)2.3.5.02	Devices and IT applications SHOULD implement the same Web services messaging, description, and security protocols (compatible and interoperable at least).	WP2, WP5, WP6
R(D1.2)2.3.5.03	Devices and IT applications MAY implement the same Web services discovery, eventing, management and reliable protocols.	WP2, WP5, WP6
R(D1.2)2.3.6.01	The System SHOULD be Low Priced	WP 3
R(D1.2)2.3.6.02	To fulfil the requirements of process automation the system MUST be Multi Functional, for the requirements of factory automation it CAN be.	WP 3
R(D1.2)2.3.6.03	The concept/architecture MUST be applicable on devices with Small Resources	WP 3
R(D1.2)2.3.6.04	The system SHOULD be able to work with a maximum degree of Autonomy	WP3
R(D1.2)2.3.6.05	The system SHOULD be able to fulfil Attribute Based Addressing functions	WP 3
R(D1.2)2.3.6.06	The system SHOULD fulfil the necessary functions for Location Awareness	WP 3
R(D1.2)2.3.6.07	The system SHOULD fulfil all/several functions of Self X	WP 3

R(D1.2)2.3.6.08	The system SHOULD fulfil all/several functions of Self-Organisation/ -Configuration	WP 3
R(D1.2)2.3.6.09	The system SHOULD fulfil all/several functions of Self-Stabilisation/-Healing	WP 3
R(D1.2)2.3.6.10	The system SHOULD fulfil all/several functions of Self-Optimisation	WP 3
R(D1.2)2.3.6.11	The system SHOULD fulfil all/several functions of Self-Protection	WP 3
R(D1.2)2.3.6.12	The system SHOULD be Self-Describing	WP 3
R(D1.2)2.3.6.13	The system SHOULD fulfil all/several functions of Self-Monitoring	WP 3
R(D1.2)2.3.6.14	The system SHOULD fulfil all/several functions of Self-Management	WP 3
R(D1.2)2.3.6.15	It MUST be possible to bridge information between wireless and wired network domains	WP3, WP4
R(D1.2)2.3.7.01	The service-centric infrastructure MUST provide a diagnostics Web service for automation devices.	WP2, WP5, WP6
R(D1.2)2.3.7.02	The service-centric infrastructure MUST provide a way for troubleshooting/debugging distributed application.	WP2, WP5, WP7
R(D1.2)2.3.8.01	The SOCRADES approach MUST support the ability to configure and reconfigure machines built from smart modules.	WP7, WP3, WP4
R(D1.2)2.3.8.02	Reuse of machine components MUST be maximized	WP7, WP8
R(D1.2)2.3.8.03	Remote assistance CAN be provided by experts to rectify machine problems	WP5, WP6, WP7
R(D1.2)2.3.8.04	SOCRADES system MUST be a vendor neutral systems/open systems	WP3, WP4, WP7
R(D1.2)2.3.8.05	SOCRADES approach SHOULD provide ability for virtual engineering	WP7
R(D1.2)2.3.8.06	System design MUST ensure design simplicity where possible	WP7, WP8
R(D1.2)2.3.8.07	Engineering SHOULD provide effective and seamless IT systems integration	WP4, WP5, WP7
R(D1.2)2.4.1.01	A minimal version of the service-centric infrastructure MUST be supported by physical devices with limited hardware resources, typically a 32 bits microprocessor, with 512Kbytes of Flash memory and 96Kbytes of RAM memory.	WP5
R(D1.2)2.4.2.01	The service-centric infrastructure MUST support high performances messaging.	WP2, WP5
R(D1.2)2.4.2.02	For real-time control, the system MUST support a wireless real-time network domain where the quality of control is guaranteed.	WP4
R(D1.2)2.4.3.01	Engineering SHOULD provide high level process description	WP2, WP6, WP7

R(D1.2)2.4.3.02	High level engineering tools SHOULD support plant layout support	WP7
R(D1.2)2.4.4.01	A system SHOULD provide provision for integrated production monitoring	WP7, WP6

4. Terms used

Abbreviation	Explanation
APL	Active Production List
ERP	Enterprise Resource Planning
DPL	Dayly Production List
DCS	Distributed Control System
FPL	Finished Production List
HMI	Human Machine Interface
I/O	Input / Output
In	Input
IP	Internet Protocol
MAS	Multi Agent System
OEE	Overall Equipment Effectiveness
Out	Output
PID	Proportional Integral Derivative controller
PDA	Personal Digital Assistent
PLC	Programmable Logical Controler
SCADA	Supervisory Control and Data Acquisition
SFC	Sequential Function Chart
SNIR	Signal to Noise and Interference Ration
SOA	Service Oriented Architecture
UC	Use Case
UDP	User Datagram Protocol
UML	Unified Markup Language
WSAN	Wireless Sensor Actuator Network
WSN	Wireless Sensor Network
XML	Extensible Markup Language

5. References

[1] Network Working Group, S. Bradner, Harvard University, Request for Comments: 2119, March 1997

[2] K. Feldmann, K,; C. Schnur and A. W. Colombo, "Modularized, Distributed Real-Time Control of Flexible Production Cells, Using Petri Nets", IFAC Journal CEP (Control Engineering Practice), vol. 4, num. 8, pp. 1067-1078, 1996.

[3] K. Feldmann and A. W. Colombo, "Material Flow and Control Sequence Specification of Flexible Production Systems Using Coloured Petri Nets", International Journal of Advanced Manufacturing Technology, 14: pp. 760-774, 1998

[4] Colombo, A. W., Neubert, R.; Schoop, R.: "A Solution to Holonic Control Systems". Proc. of the 8th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA´01), Special Session on Multi-Agent-based Factory Automation. Sophia/Nice, France, October 2001.

[5] Colombo, A. W.; Schoop, R.; Neubert, R.: "Collaborative (Agent-Based) Factory Automation". Chapter 109 in: "The Industrial Information Technology Handbook", Richard Zurawski (Ed.). CRC Press LLC, Boca Raton, USA. November 2004.

[6] Colombo, A. W.; Schoop, R.; Neubert, R.: "An Agent-based Intelligent Control Platform for Industrial Holonic Manufacturing Systems". IEEE Transaction on Industrial Electronics (IEEE-IES). February 2006.