EUROPEAN COMMISSION

Thematic Priority:
SIXTH FRAMEWORK PROGRAMME

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 [D 1.1]

State of the art - Glossary

(Final Draft)

Dissemination Level: RE

Organization name of lead contractor for this deliverable: ifak

Date: [31.03.2007]

Status description:

Scheduled completion date[1]:	31.03.2007	Actual completion date[2]:	13.04.2007
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) Emmanuel Witrant (KTH) Maben Rabi (KTH) Mikael Johansson (KTH) Karl Henrik Johansson (KTH)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)	**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 [ ] [ ] Jaguar Cars Ltd. [ ] [ ] ARM Ltd.	
Peer review approval :	[x] Approved [ ] Rejected (improve as specified hereunder)	Date:	27.03.2007
Suggested improvements:	No suggested improvements		

[1] As defined in the DoW
[2] Scheduled date for approval

Version History:

Version:	Comments, Changes, Status:	Person(s):
1.00	First released version	Thomas Bangemann (ifak) Francois Jammes (SE) Antoine Mensch (SE) Martin Strand (ABB) Oliver Baecker (SAP) Stamatis Karnouskos (SAP) Luciana Moreira Sa de Souza (SAP) Emmanuel Witrant (KTH) Maben Rabi (KTH) Mikael Johansson (KTH) Karl Henrik Johansson (KTH)Rob Harrison (Lboro) Radmehr Monfared (Lboro) Axel Klostermeyer (Siemens) Mats Lillqvist (FLEX) Axel Bepperling (SE) Jerker Delsing (LTU) Harm Smit (SE) Patrik Spiess (SAP)

List of Participants:

Number:	Participant name:	Participant short name:
1	Schneider Electric	SE
2	ABB	ABB
3	APS GmbH	APS
4	Boliden AB	BOL
5	FlexLink Automation Oy.	FLEX
6	Institut f. Automation und Kommunikation e.V. Magdeburg	ifak
7	Kungliga Tekniska Högskolan	KTH
8	Loughborough University	Lboro
9	Luleå University of Technology	LTU
10	Politecnico di Milano	PoliMi
11	SAP AG	SAP
12	Siemens AG	Siemens
13	Tampere University of Technology	TUT
14	Jaguar Cars Ltd.	JAGUAR
15	ARM Ltd.	ARM

Dissemination Level:

PU	Public
PP	Restricted to other programme participants (including the Commission Services)
RE	Restricted to a group specified by the consortium (including the Commission Services)
CO	Confidential, only for members of the consortium (including the Commission Services)

Executive Summary

The described state of the art in D1.1 and Annex to D1.1 gives an overview of technologies and aspects that enables service oriented architectures, with the main focus on networked embedded systems.

This document collects main terms of the technologies as an initial glossary. It is the intention that these terms shall be used in each of the project's work packages. Of course, this is only a small subset that will be extended during the SOCRADES project work. To this end, the terms will be transferred to a document that will be updated over the project life time. As result, the glossary will constitute a knowledge base for the project.

Table of Contents:

Glossary

Term	Explanation	Source domain
Central control	Instrumentation architecture of automation control tasks in a way that the automation algorithms are implemented in a programmable device and the I/Os are connected to this device	
Choreography	Choreography provides a formal description of peer-to-peer collaborations of parties by defining, from a global viewpoint, their common and complementary observable behaviour, i.e. ordered message exchanges. The internal behaviour of each participant is not addressed	
Cognitive agents	Cognitive agents are similar to the deliberative agents. In contrast to the reactive agents, cognitive agents possess an explicit knowledge of their environment.	
Collaborative agents	Special set of the following attributes autonomy, co-operation, learning and versatility characterise the collaborative agents	
Collaborative Automation	The umbrella paradigm for the creation of collaborative agent-based control is the so-called "collaborative automation" in which the underpinning novel approach is to consider the set of intelligent system units as a conglomerate of distributed, autonomous, intelligent, pro-active, fault-tolerant and reusable units, which operate as a set of co-operating entities.	
Commissioning	All tasks to bring a system into run	Automation common
Configuration	Network and device settings that adjust the structure of the network and/or the devices	Industrial communication
De-central control	Instrumentation architecture of automation control tasks in a way that the automation algorithms are partly implemented in field devices and in a single programmable device.	
Deliberative Agents	Deliberative agents are characterised by their goal-oriented behaviour, knowledge representation, reasoning model and the planning process that aims the generation of correct and optimal sequences of actions, having the capability of anticipation.	
Devices Profile for Web Services (DPWS)	The DPWS specification defines an architecture that distinguishes two types of services: devices (hosting services) and hosted services. Devices play an important part in the discovery and metadata exchange protocols. Hosted services provide the functional behaviour of the device and depend on their hosting device for discovery. In addition to those vendor-defined services, DPWS	

	specifies a set of built-in services: Discovery services (WS-Discovery): those services are used by a device connected to a network to advertise itself and to discover other devices. WS-Discovery uses UDP and a multicast address to broadcast and listen to the discovery messages. Metadata exchange services (WS-MetadataExchange): those services provide dynamic access to a device's hosted services and to their metadata, such as WSDL or XML Schema definitions Event publish/subscribe services (WS-Eventing): those services are extensions of vendor-defined services, and allow other devices to subscribe to asynchronous messages (events) produced by a given vendor-defined service. DPWS is built on top of the SOAP 1.2 standard, and relies on additional Web Services specifications, such as WS-Addressing and WS-Policy, to further constrain the SOAP messaging model.	
Distributed control	Instrumentation architecture of automation control tasks in a way that the automation algorithms are freely implemented in a programmable, configurable devices and devices with fixed or parameterised functionality.	
End-Point Reference (EPR)	The key abstraction underlying WS-Addressing is the EPR, which identifies a resource. It is composed of an Address and optional Reference Parameters and Metadata description.	
FCAPS	FCAPS (fault-management, configuration, accounting, performance, and security) is an acronym for a categorical model of the working objectives of network management. Read more about it: http://searchnetworking.techtarget.com/gDefinition/0,294236,sid7_gci752173,00.html	
Flowchart	Graphical language describing mixed sequential and procedural functionality	Nematron Corporation (patent #4,852,047).
Function block	Software functional unit comprising an individual, named copy of a data structure and associated operations specified by a corresponding FB type	IEC 61499/IEC 61804 FB
Grafcet	Language for programming PLCs similar to Petri Net. Also known as SFC (Sequential Function	Vendor product

	Charts)	name
Hybrid agents	Hybrid agents combine the best features of deliberative and reactive agents, achieving fast response and generation of optimal sequences of actions.	
Instruction List	Micro processor oriented language for programming PLCs	IEC 61131-3
Interface Agent	Special set of the following attributes autonomy, co-operation, learning and versatility characterise the collaborative agents	
Ladder Logic	Description of binary behaviour for control of technical artefacts. As graphical programming language part of IEC 61131-3 as Ladder Diagram (LD).	Automation common
Middleware	Middleware is defined as the infrastructure used to deploy a distributed system. Particular middleware specifications and implementations are designed to support particular distribution models.	IEEE Distributed Systems Online, Vol. 6(3), March 2005
Mobile agent	The mobile agent is able to move around in a network, interact with foreign hosts, gather information on behalf of its owner and comes back home having performed the duties set by its user.	
Multi-agent system	A multi-agent system can be defined as a set of agents that represent the objects of a system, capable to interact, in order to achieve their individual goals.	
Ontology	The term ontology is vague and not precise. [Gruber, 1995] defines ontology as a specification of a conceptualisation. [Guarino, 1998] extends the previous definition, saying that an ontology is a logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualisation of the world. Besides these differences, it is consensual that the purpose of ontologies is to create shared understanding between co-operative agents, enabling the exchange of knowledge and the capability to reuse that knowledge.	[Gruber, 1995] [Guarino, 1998]
OPC UA	The OPC Unified Architecture is the new version, based on Web services, of the well-known OPC architecture originally designed by the OPC Foundation to connect control devices to control and supervision applications.	
Orchestration	Orchestration describes processes that model the behaviour of each of the participants in an interaction involving several parties.	
Parameterisation	Setting of the variable values of devices	Automation common

Petri Net	Theory for modelling event discrete systems	Control theory
Reactive agents	**Reactive agents** do not have internal knowledge representation and operate in a stimulus-response manner aiming to produce robust actions in contrast with deliberative agents that aim to produce optimal actions. This reactive behaviour is described as situation-action rules.	
SOA Reference Model	The goal of the reference model for SOA is to define the essence of service oriented architecture, and emerge with a vocabulary and a common understanding of SOA. It provides a normative reference that remains relevant for SOA as an abstract and powerful model, irrespective of the various and inevitable technology evolutions that will influence SOA deployment.	OASIS
Sequential Function Chart	Ref. Grafcet	IEC 61131-3
Service Component Architecture	Service Component Architecture (SCA) is a set of specifications which describe a model for building applications and systems using a Service-Oriented Architecture.	OASIS
Service Component Architecture Component	SCA component offers its functionality as services and accesses other components via references.	
Service Component Architecture Composite	SCA supports the deployment of the SCA components by defining a set of XML documents containing the description of the components and their configuration.	
Service Oriented Architecture (SOA)	WP1: A service-oriented architecture (SOA) is a set of architectural tenets for building autonomous yet interoperable systems.	Service Oriented Architecture
	OASIS: Service Oriented Architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains.	OASIS
Service provider	The service provider is a software application able to receive and consume XML messages, and possibly to respond with other XML messages.	
Smart agent	Special set of the following attributes autonomy, co-operation, learning and versatility characterise	

	the collaborative agents	
Structured Text	Higher level language for programming PLCs	IEC 61131-3
Management		
Device		
Universal Plug and Play	The UPnP standard defines two main types of participants in the UPnP architecture: **devices** and **control points**. A device provides one or more **services** and acts as a server, responding to requests from control points acting as clients. Both control points and devices can be implemented on a variety of platforms including personal computers and embedded systems. A UPnP device is a container of services and, possibly, nested devices. Different categories of UPnP devices are associated with different sets of services and embedded devices. All of this information is captured in an XML device description document hosted by the device. This device description lists the set of hosted services, as well as the properties associated with the device (device name, manufacturer, model number, etc.	
Web-Service	The basic protocols and specifications to exchange structured messages between cooperating entities.	Web technology
Web-Service Discovery	WS-Discovery is based on SOAP 1.2 and builds on WS-Addressing: all discovery messages are SOAP 1.2 messages extended with WS-Addressing headers	proprietary specification
Web-Service Enumeration	WS-Enumeration defines a simple protocol for enumeration that allows a resource to provide a session abstraction, called an enumeration context, to a consumer that represents a logical cursor through a sequence of data items.	W3C member submission
Web-Service Eventing	WS-Eventing describes a protocol allowing one Web Service (called an "event sink") to register interest (called a "subscription") with another Web Service (called an "event source") in receiving messages about events (called "notifications").	W3C member submission
Web-Service Notification	The WS-Notification family of specifications defines a standard Web Services approach to notification. It encompasses the WS-BaseNotification, WS-Topics and WS-BrokeredNotification specifications.	OASIS specification
Web-Service Resource	WS Resource Framework is a collection of five specifications, which provide a richer alternative to the WS-Transfer specification:	WSRF – OASIS specification

Framework		
Web-Service Policy	WS-Policy provides a flexible and extensible grammar for expressing the capabilities, requirements, and general characteristics of entities in an XML Web services-based system.	(W3C member submission)
Web-Service Security	WS- Security proposes a standard set of SOAP extensions that can be used when building secure Web services to implement message content integrity and confidentiality.	(OASIS specification)
Web-Service Security Policy	WS-SecurityPolicy defines a set of security policy assertions which apply to Web Services Security: SOAP Message Security, WS-Trust, and WS-SecureConversation.	(OASIS member submission)
Web-Service Trust	WS-Trust uses the secure messaging mechanisms of WS-Security to define additional primitives and extensions for the issuance, exchange and validation of security tokens.	(OASIS member submission)
Web-Service Security Conversation	WS-SecureConversation defines extensions that build on WS-Security and WS-Trust to provide secure communication across one or more messages.	(OASIS member submission)
Web-Service Reliability	WS-Reliability is a SOAP-based specification that fulfils reliable messaging requirements critical to some applications of Web Services.	(OASIS specification)
Web-Service Reliable Messaging	WS-ReliableMessaging describes a protocol that allows messages to be delivered reliably between distributed applications in the presence of software component, system, or network failures.	(OASIS draft specification)
Web-Service Management	WS-Management describes a general SOAP-based protocol for managing systems such as PCs, servers, devices, Web services and other applications, and other manageable entities.	(Distributed Management Task Force - DMTF submission)
Web-Service Distributed Management	Web Services Distributed Management – WSDM specifies how the manageability of a resource is made available to manageability consumers via Web services.	(OASIS specification)
Web-Service Transfer	WS-Transfer defines a simple protocol for creating, retrieving, updating and deleting resources, which are entities addressable by an endpoint reference that provide an XML representation.	W3C member submission
Web-Service	The “WS-*” token stands for “all specifications related to Web Services that provide added-value	Web technology

Token, (WS-*)	protocols or functions on top of them". Many of these specifications are described as Web Services extensions, using the standardized "module" mechanism defined by SOAP 1.2. Consequently, these WS-* specifications can only be used with SOAP 1.2 Web Services stacks.