ISO/IEC JTC 1/WG 7
Working Group on Sensor Networks

Document Number:	N087
Date:	2010-09-06
Replace:	
Document Type:	Working Draft Text
Document Title:	1st Working Draft of ISO/IEC 20005, Information technology — Sensor Networks — Services and Interfaces Supporting Collaborative Information Processing in Intelligent Sensor Networks
Document Source:	Project Editor
Document Status:	As per the JTC 1/WG 7 Gaithersburg recommendation 4, this WD is circulated to JTC 1 NBs and WG 7 members for comment.
Action ID:	COM
Due Date:	2010-11-15
No. of Pages:	50

ISO/IEC JTC 1/WG 7 Convenor:

Dr. Yongjin Kim, Modacom Co., Ltd (Email: cap@modacom.co.kr)

ISO/IEC JTC 1/WG 7 Secretariat:

Ms. Jooran Lee, Korean Standards Association (Email: jooran@kisi.or.kr)

Reference number of working document: **ISO/IEC JTC 1/ WG7 N 087**

Date: 2010-09-06

Reference number of document: **ISO/IEC WD 20005**

Committee identification: ISO/IEC JTC 1/WG 7

Secretariat: KATS

Information technology — Sensor Networks — Services and Interfaces Supporting Collaborative Information Processing in Intelligent Sensor Networks

Document type: International Standard
Document subtype: If applicable
Document stage: (20) Preparatory stage
Document language: E

Copyright notice

Contents

Page

Foreword v
Introduction vi
1 Scope 1
2 Normative references 1
3 Terms and definitions 1
4 Abbreviations 2
5 General Description 3
5.1 Sensor networks overview 3
5.2 Requirements of intelligent sensor networks 4
5.3 Collaborative information processing overview 4
5.4 Functional model of collaborative information processing 5
5.5 Services supporting CIP Overview 6
5.5.1 Core services supporting CIP 6
5.5.2 Enhanced services supporting CIP 7
6 Core Services and Interfaces Specifications 7
6.1 Event service 7
6.1.1 EVENT-SUB.request 8
6.1.2 EVENT-SUB.indication 9
6.1.3 EVENT-SUB.confirm 10
6.1.4 EVENT-REG.indication 10
6.1.5 EVENT-UNSUB.request 11
6.1.6 EVENT-UNSUB.confirm 11
6.2 Logical grouping service 12
6.2.1 LG-ESTABLISH.request 13
6.2.2 LG-ESTABLISH.indication 13
6.2.3 LG-ESTABLISH.confirm 14
6.2.4 LG-MEMBERIN.request 14
6.2.5 LG-MEMBERIN.confirm 15
6.2.6 LG-MEMBEROUT.request 15
6.2.7 LG-MEMBEROUT.confirm 16
6.2.8 LG-DISMISS.request 16
6.2.9 LG-DISMISS.indication 17
6.2.10 LG-DISMISS.confirm 17
6.2.11 LG-QUERY.request 18
6.2.12 LG-QUERY.confirm 18
6.3 Data synchronization and registration service 19
6.3.1 SYNREG-SYNQUERY.request 20
6.3.2 SYNREG-SYNQUERY.confirm 20
6.3.3 SYNREG-SYNEXEC.request 21
6.3.4 SYNREG-SYNEXEC.indication 21
6.3.5 SYNREG-SYNEXEC.confirm 22
6.3.6 SYNREG-REGQUERY.request 22
6.3.7 SYNREG-REGQUERY.confirm 23
6.3.8 SYNREG-REGEXEC.request 24
6.3.9 SYNREG-REGEXEC.indication 24
6.3.10 SYNREG-REGEXEC.confirm 25
6.4 Information description service 25
6.4.1 INFO-LEVELGET.request 26
6.4.2 INFO-LEVELGET.confirm 27

6.4.3 INFO-LEVELSET.request 27
6.4.4 INFO-LEVELSET.indication 28
6.4.5 INFO-LEVELSET.confirm 28
6.4.6 INFO-DATA.request 29
6.4.7 INFO-DATA.indication 29
6.4.8 INFO-DATA.confirm 30
6.5 Node-to-node inter-activation service 30
6.5.1 N2NACT.request 32
6.5.2 N2NACT.confirm 32
6.6 Parameter adaptation service 33
6.6.1 PAR.request 34
6.6.2 PAR.indication 34
6.6.3 PAR.confirm 35

7 Enhanced Services and Interfaces Specifications 35
7.1 QoS management service 36
7.1.1 Service specification 36
7.1.2 Primitives and parameters 36
7.2 CIP-driven node state scheduling service 36
7.2.1 Service specification 36
7.2.2 Primitives and parameters 37
7.3 Context management service 37
7.3.1 Service specification 37
7.3.2 Primitives and parameters 37

Annex – A Core Services and Interfaces Examples 1

Annex – B Enhanced Services and Interfaces Examples 1

Annex – C Bibliography 1

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

Technical Committee ISO/IEC JTC 1, Information Technology, Working Group 7, *Sensor Networks*, prepared ISO/IEC 20005.

Introduction

Intelligent sensor networks are becoming increasingly attractive in a wide range of applications to meet generic challenges from the dynamic changes of deploying environment, network status and application performance requirement. Collaborative information processing (CIP), which closely integrates information processing algorithms with collaboration mechanisms, is an essential technology helping intelligent sensor networks to guarantee system performance in real application scenarios. This standard specifies services and interfaces supporting CIP in intelligent sensor networks.

(To be revised)

Information technology — Sensor Networks — Services and Interfaces Supporting Collaborative Information Processing in Intelligent Sensor Networks

1 Scope

This international standard specifies services and interfaces supporting collaborative information processing (CIP) in intelligent sensor networks which includes:

- CIP functionalities and CIP functional model
- Common services supporting CIP
- Common service interfaces to CIP

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ITU-T Recommendation Y.2221, *Requirements for support of Ubiquitous Sensor Network (USN) applications and services in NGN environment (2009)*

ITU-T Recommendation X.902 | ISO/IEC 10746-2 :2010, *Information technology – Open distributed processing – Reference Model: Foundations.*

ITU-T Recommendation X.903 | ISO/IEC 10746-3 :2010, *Information technology – Open distributed processing – Reference Model: Architecture.*

(To be added)

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

(Note: This clause will later be re-organized as alphabetical order of terms. And, some terms and definitions are included directly from ISO/IEC JTC1 SGSN N149, *SGSN Technical Document Version 3*)

3.1
sensor networks
A system of spatially distributed sensor nodes interacting with each other and, depending on applications, interacting with ICT infrastructures, in order to acquire, process, transfer, and provide information from the physical world and optionally react.

3.2
sensor node
A sensor node is a device that consists of at least one sensor and zero or more actuators, and has processing and networking capabilities through wired or wireless means. [ISO/IEC JTC1 SGSN N149]

3.3
Sensor network service
A structural set of capabilities or functions which are offered by the sensor nodes or sensor networks.

3.4
service set or service subset
A group or subgroup of services organized to provide common mechanisms or facilities to meet certain requirements from users or applications.

3.5
sensor network application
The sensor network application is a user case of sensor networks supporting a set of sensor network services for users. [ISO/IEC JTC1 SGSN N149]

3.6
information
Any kind of knowledge that is exchangeable amongst users or entities, about things, facts, concepts and so on, in a universe of discourse.

3.7
information processing
The manipulation of data or information so that new data or information which is implicit in the original be appeared in a useful form, or with which further information processing can be applied and/or be utilized to make a response suitable within the context of an objective, problem or situation.

3.8
collaborative information processing
A form of information processing in which multiple discrete components or entities participate in a manner of collaboration, in order to enhance processing efficiency and to improve quality and reliability of the results.

3.9
viewpoint
A form of abstraction achieved using a selected set of architectural concepts and structuring rules, in order to focus on particular concerns within a system. [ISO/IEC 10746]

4 Abbreviations

For the purposes of this document, the following abbreviations apply.

QoS	Quality of Service
CIP	Collaborative Information Processing
SAP	Service Access Point
CDE	Capability Declaration Entity
CSPE	Collaborative Strategy Planning Entity
CRSE	Communication Requirement Specification Entity
OSI	Open System Interconnection

GSR Generalized System Requirement

FCR Functional Capability Requirement

(to be added)

5 General Description

Sensor networks have been widely deployed in different application domains ranging from environment monitoring, transportation, industries, and healthcare etc. Wired/wireless sensor networks can be regarded as an extension of Internet towards the physical world. In order to meet challenges from intrinsic environment complexity, large orders of magnitude network scaling and dynamic application requirements, intelligent sensor networks are developed to provide new system capabilities such as environment self-adaptability, dynamic task supporting and autonomous system maintenance. Collaborative information processing (CIP) is an essential technology to implement new functionalities of intelligent sensor network and finally to provide improved services to users of intelligent sensor networks.

This clause gives general description of sensor networks and requirements of intelligent sensor networks. Overviews of CIP and its functional models are presented, followed by general explanation on core and enhanced services supporting CIP in intelligent sensor networks.

5.1 Sensor networks overview

Compared with the traditional networks, sensor networks not only provide information transmission service but also provide information sensing, processing, provision and other services. Figure 1 shows a conceptual overview of the functional architecture of sensor networks.

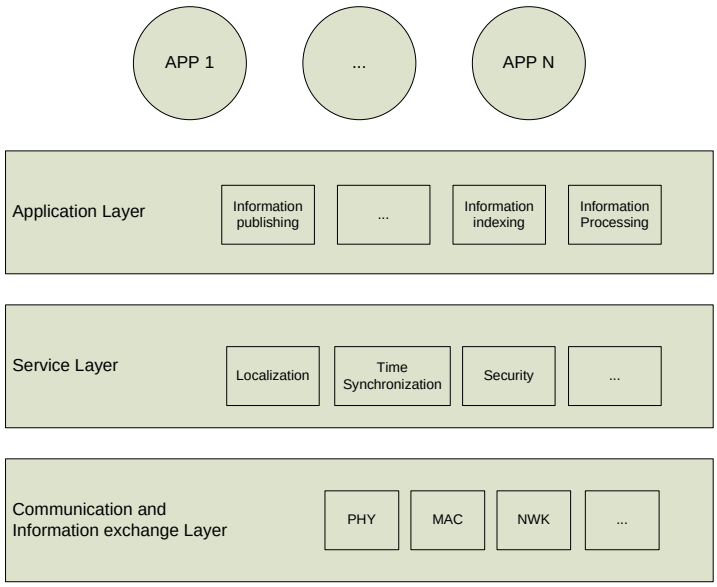

Figure 1 — Overview of the functional architecture of sensor networks

Communication and information exchange layer implements functionalities fulfilled by the lower layers in the OSI stack, including physical layer, data link layer, network layer, and transport layer. Application layer provides services to individual applications and implements functions such as information publishing, information indexing and information processing etc. Between application layer and communication and information exchange layer, the service layer provides generic common services to entities in the above application layer. In the context of sensor networks, a lot of generic common services need to be implemented including localization service, time synchronization service, security service and other services.

5.2 Requirements of intelligent sensor networks

Besides the generalized system requirements (GSR) and generalized functional capability requirements (FCR) of sensor networks, there are additional unique requirements on intelligent sensor networks to meet challenges from the dynamic changes of deploying environment, network status and application performance requirement.

- **Environmental self-adaptability:** Intelligent sensor network shall adapt to obtain required system performances if the physical environment changes. As an example, an intelligent sensor network based anti-intrusion system should guarantee consistent system performance such as false alarming rate when the environment in which the network is deployed changes.

- **Dynamic task supporting:** Intelligent sensor network shall support dynamic tasks including dynamic task assigning, dynamic service-providing and dynamic quality of service (QoS).

- **Autonomous system maintenance:** Intelligent sensor network can autonomously maintain system functionalities in case of network scaling, node mobility and node failures.

5.3 Collaborative information processing overview

The key difference between traditional telecommunication infrastructures and sensor networks based information service system is that sensor networks based information service system collects low-level sensory data, extracts application-specific information from these sensory data, and make attempt to obtain high-level data, information, and knowledge about physical world.

Integrated with other entities such as sensory information description, sensor identification and sensory information storage, CIP concerns on how to resource-efficiently fulfil dynamic tasks specified by information service consumer. Though different sensor network application scenarios normally require scenario-specific services, collaboration is an indispensable requirement for senor network based information service to handle constraints in energy, computing, storage and communication bandwidth. To information service provider, it also has to deal with technical challenges from issues such as task dynamics, measurement uncertainty, node mobility and environmental changing.

The aim of CIP in sensor networks is to improve system efficiency, enhance quality of service and guarantee system performance. It provides efficient mechanisms and/or protocols to meet generic challenges from the dynamic changes of deploying environment, network status and application performance requirement.

CIP can be viewed from three distinct viewpoints. Figure 2 shows a three-dimensional functional model of CIP.

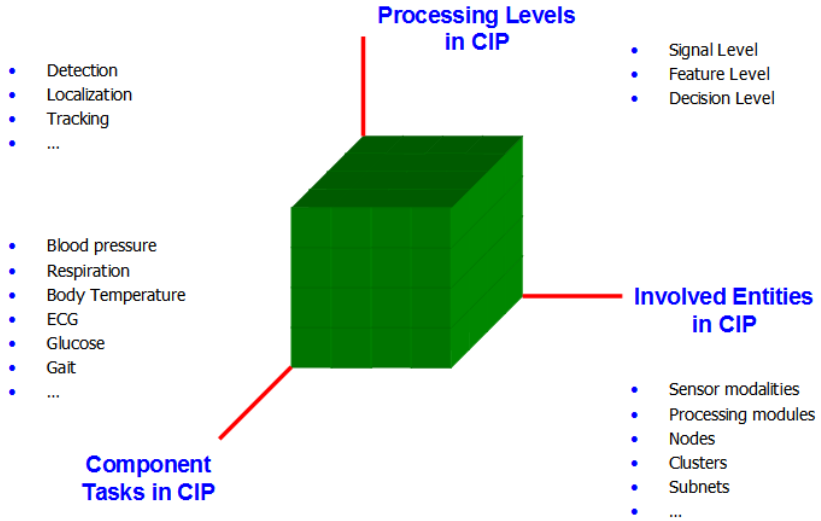

Figure 2 — Functional model of collaborative information processing

The first viewpoint is Processing Level viewpoint. In this viewpoint, CIP can be implemented on different processing levels, which includes data, feature and decision processing levels. The second viewpoint is Involved Entity. Involved Entities in CIP could be sensor modalities, processing modules, nodes, clusters, and even subnets. CIP can thirdly be viewed from Component Task perspective. Elements in this viewpoint depend on the specific application scenarios of sensor networks. In healthcare context, component tasks may include blood pressure/temperature measurement, respiration inspection, and gait analysis. In an anti-intrusion application, target detection, classification, and tracking could be component tasks for security services. Specific selections and combinations using elements from these three dimensions correspond to different application task implementations, or personalized services.

5.4 Functional model of collaborative information processing

Figure 3 shows a functional model of collaborative Information Processing. In this model, CIP can be characterized by three distinct entities, which is named as capability declaration entity (CDE), collaborative strategy planning entity (CSPE) and communication requirement specification entity (CRSE).

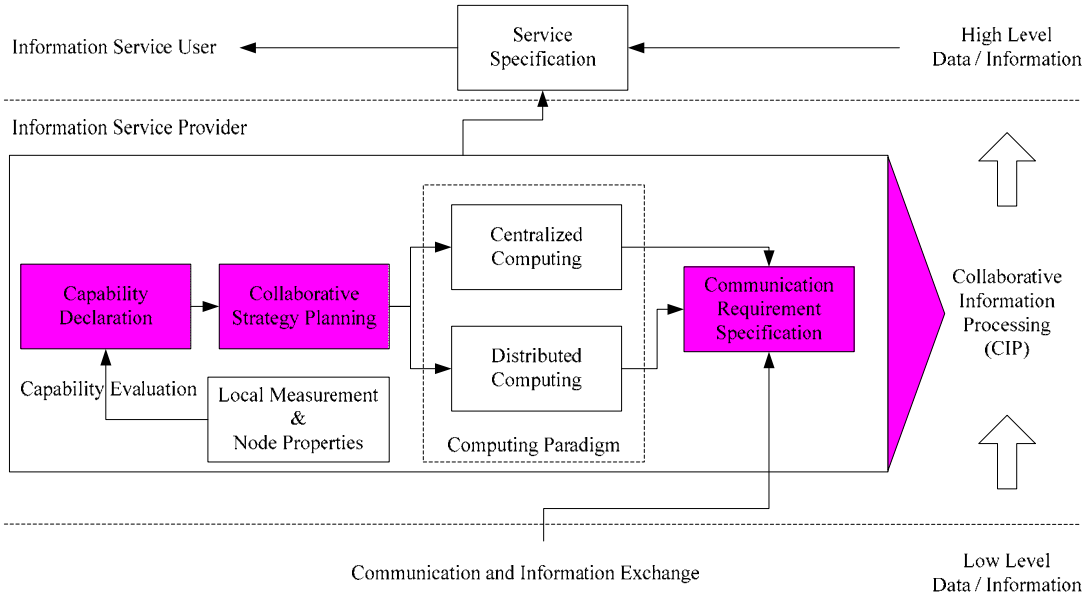

Figure 3 — Functional model of collaborative information processing

Capability declaration entity (CDE) declares capabilities of one sensor node to other nodes. Capabilities include not only individual node information on sensing modality configuration, sensing range, residual energy, storage and communication bandwidth etc., but also include certain characteristic information of sensory data collected by individual sensor node. One of the representative characteristics on sensory data is signal-to-noise ratio (SNR) value. Other characteristics include signal energy, estimated distance from target and sensor nodes, and state parameter prediction, etc. In other words, one sensor node should qualify itself to be a CIP participant before any actual CIP procedure is triggered. CDE requires a preliminary local capability evaluation process which uses information of local measurement and node property.

Collaborative strategy planning entity (CSPE) is the second and probably the most important entity in CIP. CSPE uses available information provided by CDE and forms global or regional maps or scopes on signal and information processing problems. With certain cost functions or utility measures, CSPE tries to find a resource-efficient solution to collaborative strategy planning problem, with which the best information processing performance can be achieved at the same time. Two computing paradigms can be used in the implementation of resulting solution from CSPE. One is centralized computing paradigm; the other is distributed computing paradigm.

Communication requirement specification entity (CRSE) acts as interface between information service provider and Communication and Information Exchange. CRSE defines parameters, languages or protocols to clearly describe requirements on communication and information exchange. Different requirements, such as end-to-end delay, time jitter, bit error and other QoS parameters should be specified.

5.5 Services supporting CIP Overview

A lot of generic common services can be provided by the service layer as shown in Figure 1. In the context of intelligent sensor networks, this standard specifies a subset of these generic common services which interface with CIP entities in the application layer and support implementation of corresponding CIP entity functionalities.

Services supporting CIP can be conceptually divided into two classes: core services and enhanced services. Core services include fundamental and essential services which can be provided directly and individually to CIP entities. Enhance services is implemented through service combination and integration of two or more core services or other generic common services provided by the service layer.

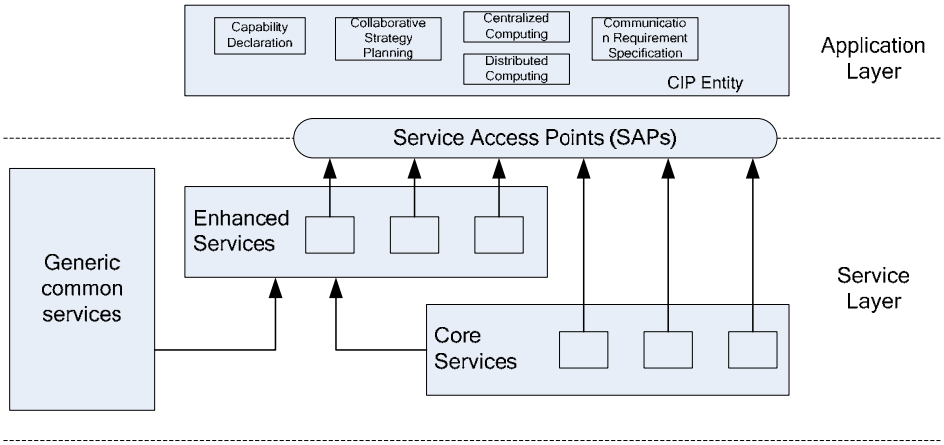

Figure 4 —Overview of services supporting CIP

5.5.1 Core services supporting CIP

The core services supporting CIP includes:

- **Event service:** This service implements functionalities concerning on the process of event subscription, registration, cancellation and un-subscription. Event may be generated due to environmental change, new physical signal occurrence and network status dynamics.

- **Logical grouping service:** This service implements functionalities concerning on the process of establishment and management of logical group for the implementation of CIP in the application layer. Logical grouping service provides mechanism for establishing collaborative relationship among entities in intelligent sensor network.

- **Data synchronization and registration service:** This service implements functionalities to synchronize different types of data sources and provides registration functions.

- **Information Description service:** This service provides mechanisms to establish ways or methods to describe information in intelligent sensor network. Information can be the input parameters to CIP processes, and it can also be the output results from CIP processes

- **Node-to-node inter-activation service:** This service provides mechanisms to trigger or initiate tasks executing in one node by another node. Dynamic tasking can be supported by this core service.

- **Parameter adaptation service:** This service provides mechanisms to adapt or reconfigure parameters for CIP. Parameter adaptation service is one of the essential services to guarantee system performance in case of dynamic changes of deploying environment and application requirement.

(to be further revised. Contributions are requested.)

5.5.2 Enhanced services supporting CIP

The enhanced services supporting CIP includes:

- **QoS management service:** This service provides mechanisms to describe QoS metrics or levels, to apply QoS rules, and to configure QoS profiles. QoS management in intelligent sensor network should be implemented from both information processing perspective and communication processing perspective.

- **CIP-driven node state scheduling service:** This service provides functions to control and schedule node states according to the request of CIP entities in stead of node management entities in intelligent sensor networks. This service can help to implement application-oriented networking and on demand task scheduling.

- **Context management service:** This service provides mechanisms to describe, to discover and to manage context information in intelligent sensor network. In case of node mobility, context management service can provide autonomous system maintenance in intelligent sensor network.

(to be further revised. Contributions are requested.)

6 Core Services and Interfaces Specifications

This clause specifies core services supporting CIP in intelligent sensor networks. Service primitives and parameters of primitives are defined for each core service. In Table 1, the names of service access points (SAPs) through which specific service is provided are listed.

Table 1 — Core services and the names of SAPs

Service name	SAP name
Event service	EVENT-SAP
Logical grouping service	LG-SAP
Data synchronization and registration service	SYNREG-SAP
Information description service	INFO-SAP
Node-to-node Inter-activation service	N2NACT-SAP
Parameter adaptation service	PAR-SAP

6.1 Event service

In this subclause, event service in the service layer is defined. Event service is provided through EVENT-SAP. The EVENT-SAP is the logical interface between event service entity in the service layer and CIP entity in the application layer. This logical interface incorporates a set of primitives and their definitions. These primitives and definitions are described conceptually here, but through this the process of the parameters exchanged

between the service layer and the application layer can be understood. Table 2 lists the primitives supported by the EVENT-SAP. Table 3 outlines primitive parameters.

Table 2 — EVENT-SAP primitive summary

Name	Request	Indication	Response	Confirm
EVENT-SUB	6.1.1	6.1.2		6.1.3
EVENT-REG		6.1.4		
EVENT-UNSUB	6.1.5			6.1.6

Table 3 — EVENT-SAP primitive parameters

Name	Type	Valid range	Description
EVSubmodel	Enumeration	CHANNEL, TYPE, FILTER, GROUP	Event subscription models.
EVSubValue	Various	Subscription model specific	Value of event attribute under specific subscription model.
EVSubSourceID	Integer		Source of event subscription
EVSubDestinationID	integer		Destination of event subscription
EVSubResultCode	Enumeration	SUCCESS, INVALID_MODEL, INVALID_VALUE, FAILED	The result of the event subscription.
EV_Time	Integer	Implementation specific	A time indication for the event occurrence, as provided by the service layer.

6.1.1 EVENT-SUB.request

This primitive requests the process of event subscription from the application layer.

6.1.1.1 Definition of service primitives

The syntax of this primitive is:

EVENT-SUB.request {

EVSubmodel,

EVSubValue,

EVSubSourceID,

EVSubDestinationID

}

Table 3 defines the parameters of this primitive.

6.1.1.2 When generated

This occurs by CIP entity to subscribe event service from the entity of the service layer.

6.1.1.3 Effect of receipt

On receipt of this primitive, the entity providing the event service implements event subscription in the EVSubDestinationID node for the EVSubSourceID node. Four types of event subscription models are provided: CHANNEL, TYPE, FILTER and GROUP.

In CHANNEL event subscription model, event channels are like television channels – if you view a channel, you receive all program broadcast on that channel. Subscribers in this model may listen to several channels. On the other hand, an event is not necessarily associated with one specific channel – it may be distributed over several channels. In TYPE event subscription model, events are filtered based upon their types. When a subscriber is only interested in a subset of all notifications available at a producer, FILTER event subscription model can be used. The subscriber defines the filter criteria which are checked by the producer. Subscribers using the same filter criteria can be grouped together. A producer in GROUP event subscription model delivers event occurrences to the whole group.

6.1.2 EVENT-SUB.indication

This primitive indicates the event subscription from the service layer to CIP entity.

6.1.2.1 Definition of service primitives

The syntax of this primitive is:

EVENT-SUB.indication {

EVSubmodel,

EVSubValue,

EVSubSourceID,

EVSubDestinationID,

}

Table 3 defines the parameters of this primitive.

6.1.2.2 When generated

This occurs when the service layer indicates a result of event subscription to CIP entity.

6.1.2.3 Effect of receipt

On receipt of this primitive, the CIP entity is indicated the result of event subscription.

6.1.3 EVENT-SUB.confirm

This primitive confirms an event subscription from the service layer to the CIP entity.

6.1.3.1 Definition of service primitives

The syntax of this primitive is:

EVENT-SUB.confirm {

EVSubSourceID,

EVSubDestinationID,

EVSubResultCode

}

Table 3 defines the parameters of this primitive.

6.1.3.2 When generated

This primitive reports a result of event subscription request.

6.1.3.3 Effect of receipt

If the EVSubResultCode is SUCCESS, it means the event subscription is successful, otherwise an error is indicated.

6.1.4 EVENT-REG.indication

This primitive indicates the event occurrence from the service layer to CIP entity.

6.1.4.1 Definition of service primitives

The syntax of this primitive is:

EVENT-REG.indication {

EVSubSourceID,

EVSubDestinationID,

EV_Time

}

Table 3 defines the parameters of this primitive.

6.1.4.2 When generated

This occurs when the service layer indicates event occurrences to CIP entity. If one or more events occur or are detected, this primitive indication can be generated more than once.

6.1.4.3 Effect of receipt

On receipt of this primitive, the CIP entity is indicated the occurrence of events. The time when event occurs or is detected is provided by EV_Time.

6.1.5 EVENT-UNSUB.request

This primitive requests the cancellation of event subscription from the application layer.

6.1.5.1 Definition of service primitives

The syntax of this primitive is:

EVENT-UNSUB.request {

EVSubmodel,

EVSubValue,

EVSubSourceID,

EVSubDestinationID

}

Table 3 defines the parameters of this primitive.

6.1.5.2 When generated

This occurs by CIP entity to unsubscribe event service from the entity of the service layer.

6.1.5.3 Effect of receipt

On receipt of this primitive, the entity providing the event service cancels event subscription in the EVSubDestinationID node for the EVSubSourceID node.

6.1.6 EVENT-UNSUB.confirm

This primitive confirms an event subscription cancellation from the service layer to the CIP entity.

6.1.6.1 Definition of service primitives

The syntax of this primitive is:

EVENT-UNSUB.confirm {

EVSubSourceID,

EVSubDestinationID,

EVSubResultCode

}

Table 3 defines the parameters of this primitive.

6.1.6.2 When generated

This primitive confirms cancellation of event subscription.

6.1.6.3 Effect of receipt

If the EVSubResultCode is SUCCESS, it means the event subscription cancellation is successful, otherwise an error is indicated.

6.2 Logical grouping service

In this subclause, logical grouping service in the service layer is defined. Logical grouping service is provided through LG-SAP. The LG-SAP is the logical interface between event service entity in the service layer and CIP entity in the application layer. This logical interface incorporates a set of primitives and their definitions. These primitives and definitions are described conceptually here, but through this the process of the parameters exchanged between the service layer and the application layer can be understood. Table 4 lists the primitives supported by the LG-SAP. Table 5 outlines primitive parameters.

Table 4 — LG-SAP primitive summary

Name	Request	Indication	Response	Confirm
LG-ESTABLISH	6.2.1	6.2.2		6.2.3
LG-MEMBERIN	6.2.4			6.2.5
LG-MEMBEROUT	6.2.6			6.2.7
LG-DISMISS	6.2.8	6.2.9		6.2.10
LG-QUERY	6.2.11			6.2.12

Table 5 — LG-SAP primitive parameters

Name	Type	Valid range	Description
LGRequestorID	integer		Requestor node ID of logical grouping
LGCoordinatorID	integer		Coordinator node ID of logical group.
LGMaxNum	integer	0x00-0xff	The maximum number of logical group members
LGMemberINID	integer		Member ID that requests joining in a logical group.
LGMemberOUTID	integer		Member ID that requests quitting a logical group.

LGAttributeName	string		Attribute name of a logical group.
LGAttributeValue	Variable		Specific attribute value.
LGResultCode	Enumeration	SUCCESS, FAILED, MEMBER_NUM_OVERFLOW INVALID_LG_ATTRIBUTE	Result code of logical grouping.

6.2.1 LG-ESTABLISH.request

This primitive requests the establishment of a logical group from the application layer.

6.2.1.1 Definition of service primitives

The syntax of this primitive is:

LG-ESTABLISH.request {

LGRequestorID,

LGCoordinatorID,

LGMaxNum

}

Table 5 defines the parameters of this primitive.

6.2.1.2 When generated

This occurs by CIP entity from the application layer to request establishment of a logical group.

6.2.1.3 Effect of receipt

On receipt of this primitive, the LGCoordinatorID node establishes a logical group and declares itself as the coordinator of the new logical group. The LGCoordinatorID is used as the name or identifier of the new logical group. A logical group membership table with up to LGMaxNum entries is established and hence maintained within the LGCoordinatorID. A node can only be coordinator of no more than one logical group, but it can be simultaneously a member of multiple logical groups.

6.2.2 LG-ESTABLISH.indication

This primitive indicates the logical group establishment from the service layer to the local CIP entity.

6.2.2.1 Definition of service primitives

The syntax of this primitive is:

LG-ESTABLISH.indication {

LGRequestorID,

LGCoordinatorID,

}

Table 5 defines the parameters of this primitive.

6.2.2.2 When generated

This occurs when the service layer indicates establishment of a logical group to CIP entity.

6.2.2.3 Effect of receipt

On receipt of this primitive, the CIP entity is indicated the establishment of a logical group and hence can query attributes of this logical group.

6.2.3 LG-ESTABLISH.confirm

This primitive confirms a logical group establishment from the service layer to the CIP entity in the application layer.

6.2.3.1 Definition of service primitives

The syntax of this primitive is:

LG-ESTABLISH.confirm {

LGRequestorID,

LGCoordinatorID,

LGResultCode

}

Table 5 defines the parameters of this primitive.

6.2.3.2 When generated

This primitive reports a result of logical group establishment request.

6.2.3.3 Effect of receipt

If the LGResultCode is SUCCESS, it means a logical group coordinated by the LGCoordinatorID node is successful. If the LGResultCode is FAILED, an error is indicated to the LGRequestorID node.

6.2.4 LG-MEMBERIN.request

This primitive requests a membership of a logical group from the application layer.

6.2.4.1 Definition of service primitives

The syntax of this primitive is:

LG-MEMBERIN.request {

LGCoordinatorID,

LGMemberINID

}

Table 5 defines the parameters of this primitive.

6.2.4.2 When generated

This occurs by CIP entity from the application layer to request a membership of a logical group which is coordinated by the LGCoordinatorID node.

6.2.4.3 Effect of receipt

On receipt of this primitive, if the current member number is less than the maximum member number of the logical group, the LGCoordinatorID node adds the LGMemberINID into the membership table of the logical group. Otherwise, a LGResultCode value is generated to indicate a MEMBER_NUM_OVERFLOW error.

6.2.5 LG-MEMBERIN.confirm

This primitive confirms result of the request of a membership of a logical group to CIP entity in the application layer.

6.2.5.1 Definition of service primitives

The syntax of this primitive is:

LG-MEMBERIN.confirm {

LGCoordinatorID,

LGMemberINID,

LGResultCode

}

Table 5 defines the parameters of this primitive.

6.2.5.2 When generated

This primitive reports a result of membership request to a logical group.

6.2.5.3 Effect of receipt

If the LGResultCode is SUCCESS, it means that the LGMemberINID node successfully joins a logical group coordinated by the LGCoordinatorID node. If the LGResultCode is MEMBER_NUM_OVERFLOW, the membership request is not successful due to the constraint of the maximum member number. Otherwise, a FAILED error is confirmed.

6.2.6 LG-MEMBEROUT.request

This primitive requests a membership cancellation of a logical group from the application layer.

6.2.6.1 Definition of service primitives

The syntax of this primitive is:

LG-MEMBEROUT.request {

LGCoordinatorID,

LGMemberOUTID

}

Table 5 defines the parameters of this primitive.

6.2.6.2 When generated

This occurs by CIP entity from the application layer to request a membership cancellation of a logical group which is coordinated by the LGCoordinatorID node.

6.2.6.3 Effect of receipt

On receipt of this primitive, the LGCoordinatorID node deletes the LGMemberINID from the membership table of the logical group. Correspondingly, the current member number is decreased by 1.

6.2.7 LG-MEMBEROUT.confirm

This primitive confirms result of the request of a membership cancellation of a logical group to CIP entity in the application layer.

6.2.7.1 Definition of service primitives

The syntax of this primitive is:

LG-MEMBERIN.confirm {

LGCoordinatorID,

LGMemberOUTID,

LGResultCode

}

Table 5 defines the parameters of this primitive.

6.2.7.2 When generated

This primitive reports a result of membership cancellation request to a logical group.

6.2.7.3 Effect of receipt

If the LGResultCode is SUCCESS, it means that the LGMemberINID node successfully quits a logical group coordinated by the LGCoordinatorID node. Otherwise, a FAILED error is confirmed.

6.2.8 LG-DISMISS.request

This primitive requests the dismissal of a logical group from the application layer.

6.2.8.1 Definition of service primitives

The syntax of this primitive is:

LG-DISMISS.request {

LGRequestorID,

LGCoordinatorID,

}

Table 5 defines the parameters of this primitive.

6.2.8.2 When generated

This occurs by CIP entity from the application layer to request dismissal of a logical group.

6.2.8.3 Effect of receipt

On receipt of this primitive, the LGCoordinatorID node frees memory for the membership table and the attribute variables of current logical group. Once the operation is successful, the LGCoordinatorID node flags itself as a non-coordinator node and hence can be acted as coordinator upon a request of new group establishment. The memberships of this node to other multiple logical groups are not affected.

6.2.9 LG-DISMISS.indication

This primitive indicates the logical group dismissal from the service layer to the local CIP entity.

6.2.9.1 Definition of service primitives

The syntax of this primitive is:

LG-DISMISS.indication {

LGRequestorID,

LGCoordinatorID,

}

Table 5 defines the parameters of this primitive.

6.2.9.2 When generated

This occurs when the service layer indicates dismissal of a logical group to CIP entity.

6.2.9.3 Effect of receipt

On receipt of this primitive, the CIP entity is indicated the dismissal of a logical group.

6.2.10 LG-DISMISS.confirm

This primitive confirms a logical group dismissal from the service layer to the CIP entity.

6.2.10.1 Definition of service primitives

The syntax of this primitive is:

LG-DISMISS.confirm {

LGRequestorID,

LGCoordinatorID,

LGResultCode

}

Table 5 defines the parameters of this primitive.

6.2.10.2 When generated

This primitive reports a result of logical group dismissal request.

6.2.10.3 Effect of receipt

If the LGResultCode is SUCCESS, it means a logical group coordinated by the LGCoordinatorID node has now be successfully dismissed. If the LGResultCode is FAILED, an error is indicated to the LGRequestorID node.

6.2.11 LG-QUERY.request

This primitive queries logical group attributes by CIP entities in the application layer.

6.2.11.1 Definition of service primitives

The syntax of this primitive is:

LG-QUERY.request {

LGCoordinatorID,

LGAttributeName

}

Table 5 defines the parameters of this primitive.

6.2.11.2 When generated

This occurs by CIP entity from the application layer to query an attribute of a logical group which is coordinated by the LGCoordinatorID node.

6.2.11.3 Effect of receipt

On receipt of this primitive, the LGCoordinatorID node queries the LGAttributeName attribute of current logical group. Specific attribute values are generated.

6.2.12 LG-QUERY.confirm

This primitive returns the results of logical group attributes to CIP entity in the application layer.

6.2.12.1 Definition of service primitives

The syntax of this primitive is:

LG-MEMBERIN.confirm {

LGCoordinatorID,

LGAttributeName,

LGAttributeValue,

LGResultCode

}

Table 5 defines the parameters of this primitive.

6.2.12.2 When generated

This primitive returns the result of querying attribute of a logical group.

6.2.12.3 Effect of receipt

If the LGResultCode is SUCCESS, it means that the value of the LGAttributeName attribute is successfully returned in the LGAttributeValue parameter. Otherwise, a FAILED error is confirmed.

6.3 Data synchronization and registration service

In this subclause, data synchronization and registration service in the service layer is defined. Data synchronization and registration service is provided through SYNREG-SAP. The SYNREG-SAP is the logical interface between event service entity in the service layer and CIP entity in the application layer. This logical interface incorporates a set of primitives and their definitions. These primitives and definitions are described conceptually here, but through this the process of the parameters exchanged between the service layer and the application layer can be understood. Table 6 lists the primitives supported by the SYNREG-SAP. Table 7 outlines primitive parameters.

Table 6 — SYNREG-SAP primitive summary

Name	Request	Indication	Response	Confirm
SYNREG-SYNQUERY	6.3.1			6.3.2
SYNREG-SYNEXEC	6.3.3	6.3.4		6.3.5
SYNREG-REGQUERY	6.3.6			6.3.7
SYNREG-REGEXEC	6.3.8	6.3.9		6.3.10

Table 7 — SYNREG-SAP primitive parameters

Name	Type	Valid range	Description
SYNREGSrcID	integer		Source node ID of data synchronization and registration service.
SYNREGDstID	integer		Destination node ID of data synchronization and registration service.
SYNTimeRef	integer	Implementation specific	Value of time reference in data synchronization

			process.
SYNExecVal	integer	Implementation specific	Value for data synchronization process.
RegRefName	string		Reference attribute name in data registration process.
RegRefValue	variable		Reference value in data registration process.
REGExecVal	variable		Value for data registration process
SYNREGResultCode	Enumeration	SUCCESS, INVALID_REF_NAME, FAILED	Result code of data synchronization and registration operation.

6.3.1 SYNREG-SYNQUERY.request

This primitive queries values of reference time for data synchronization by CIP entities in the application layer.

6.3.1.1 Definition of service primitives

The syntax of this primitive is:

SYNREG-SYNQUERY.request {

SYNREGSrcID,

SYNREGDstID,

}

Table 7 defines the parameters of this primitive.

6.3.1.2 When generated

This occurs by CIP entity from the application layer to query value of reference time from the SYNREGDstID node which is required by data synchronization process in the SYNREGSrcID node.

6.3.1.3 Effect of receipt

On receipt of this primitive, the SYNREGDstID node gets current reference time for generating sensory data.

6.3.2 SYNREG-SYNQUERY.confirm

This primitive returns the values of reference time to CIP entity in the application layer.

6.3.2.1 Definition of service primitives

The syntax of this primitive is:

SYNREG-SYNQUERY.confirm {

SYNREGSrcID,

SYNREGDstID,

SYNTimeRef,

SYNREGResultCode

}

Table 7 defines the parameters of this primitive.

6.3.2.2 When generated

This primitive returns the result of querying values of reference time.

6.3.2.3 Effect of receipt

If the SYNREGResultCode is SUCCESS, it means that the value of reference time for generating sensory data is successfully returned in the SYNTimeRef parameter. Otherwise, a FAILED error is confirmed.

6.3.3 SYNREG-SYNEXEC.request

This primitive requests the execution of data synchronization by CIP entity in the application layer.

6.3.3.1 Definition of service primitives

The syntax of this primitive is:

SYNREG-SYNEXEC.request {

SYNREGSrcID,

SYNREGDstID,

SYNExecVal

}

Table 7 defines the parameters of this primitive.

6.3.3.2 When generated

This occurs by CIP entity from the application layer to request the execution of data synchronization in the SYNREGDstID node.

6.3.3.3 Effect of receipt

On receipt of this primitive, the SYNREGDstID node executes data synchronization process locally, in which SYNExecVal is used to synchronize local reference time for generating sensory data with the reference time of the SYNREGSrcID node.

6.3.4 SYNREG-SYNEXEC.indication

This primitive indicates the execution of data synchronization process from the service layer to the local CIP entity.

6.3.4.1 Definition of service primitives

The syntax of this primitive is:

SYNREG-SYNEXEC.indication {

SYNREGSrcID,

SYNREGDstID

}

Table 7 defines the parameters of this primitive.

6.3.4.2 When generated

This occurs when the service layer indicates the execution of data synchronization process to CIP entity.

6.3.4.3 Effect of receipt

On receipt of this primitive, the CIP entity is indicated the execution of data synchronization process.

6.3.5 SYNREG-SYNEXEC.confirm

This primitive confirms the execution of data synchronization from the service layer to the CIP entity.

6.3.5.1 Definition of service primitives

The syntax of this primitive is:

SYNREG-SYNEXEC.confirm {

SYNREGSrcID,

SYNREGDstID,

SYNREGResultCode

}

Table 7 defines the parameters of this primitive.

6.3.5.2 When generated

This primitive reports execution result of data synchronization process to the CIP entity.

6.3.5.3 Effect of receipt

If the SYNREGResultCode is SUCCESS, it means that the reference time for generating sensory data in the SYNREGDstID node is successfully synchronized with the reference time in the SYNREGSrcID node. If the SYNREGResultCode is FAILED, an error is indicated to the SYNREGSrcID node.

6.3.6 SYNREG-REGQUERY.request

This primitive queries reference attributes for data registration process by CIP entities in the application layer.

6.3.6.1 Definition of service primitives

The syntax of this primitive is:

SYNREG-REGQUERY.request {

SYNREGSrcID,

SYNREGDstID,

RegRefName

}

Table 7 defines the parameters of this primitive.

6.3.6.2 When generated

This occurs by CIP entity from the application layer to query value of the RegRefName attribute from the SYNREGDstID node which is required by data registration process in the SYNREGSrcID node.

6.3.6.3 Effect of receipt

On receipt of this primitive, the SYNREGDstID node gets the value of the RegRefName attribute in the SYNREGDstID node.

6.3.7 SYNREG-REGQUERY.confirm

This primitive returns value of the RegRefName attribute to CIP entity in the application layer.

6.3.7.1 Definition of service primitives

The syntax of this primitive is:

SYNREG-REGQUERY.confirm {

SYNREGSrcID,

SYNREGDstID,

RegRefName,

RegRefValue,

SYNREGResultCode

}

Table 7 defines the parameters of this primitive.

6.3.7.2 When generated

This primitive returns the result of the RegRefName attribute value in the SYNREGDstID node.

6.3.7.3 Effect of receipt

If the SYNREGResultCode is SUCCESS, it means that the value of the RegRefName attribute is successfully returned in the RegRefValue parameter. If the SYNREGResultCode is INVALID_REF_NAME, it means that the RegRefName attribute is not valid in the SYNREGDstID node. Otherwise, a FAILED error is confirmed.

6.3.8 SYNREG-REGEXEC.request

This primitive requests the execution of data registration by CIP entity in the application layer.

6.3.8.1 Definition of service primitives

The syntax of this primitive is:

SYNREG-REGEXEC.request {

SYNREGSrcID,

SYNREGDstID,

REGExecVal

}

Table 7 defines the parameters of this primitive.

6.3.8.2 When generated

This occurs by CIP entity from the application layer to request the execution of data registration in the SYNREGDstID node.

6.3.8.3 Effect of receipt

On receipt of this primitive, the SYNREGDstID node executes data registration process locally, in which REGExecVal is used to perform data registration in the SYNREGSrcID node.

6.3.9 SYNREG-REGEXEC.indication

This primitive indicates the execution of data registration process from the service layer to the local CIP entity.

6.3.9.1 Definition of service primitives

The syntax of this primitive is:

SYNREG-REGEXEC.indication {

SYNREGSrcID,

SYNREGDstID

}

Table 7 defines the parameters of this primitive.

6.3.9.2 When generated

This occurs when the service layer indicates the execution of data registration process to CIP entity.

6.3.9.3 Effect of receipt

On receipt of this primitive, the CIP entity is indicated the execution of data registration process.

6.3.10 SYNREG-REGEXEC.confirm

This primitive confirms the execution of data registration from the service layer to the CIP entity.

6.3.10.1 Definition of service primitives

The syntax of this primitive is:

SYNREG-REGEXEC.confirm {

SYNREGSrcID,

SYNREGDstID,

SYNREGResultCode

}

Table 7 defines the parameters of this primitive.

6.3.10.2 When generated

This primitive reports execution result of data registration process to the CIP entity.

6.3.10.3 Effect of receipt

If the SYNREGResultCode is SUCCESS, it means that the data registration process is successfully executed within the SYNREGDstID node. If the SYNREGResultCode is FAILED, an error is indicated to the SYNREGSrcID node.

6.4 Information description service

In this subclause, information description service in the service layer is defined. Information description service is provided through INFO-SAP. The INFO-SAP is the logical interface between information description service entity in the service layer and CIP entity in the application layer. This logical interface incorporates a set of primitives and their definitions. These primitives and definitions are described conceptually here, but through this the process of the parameters exchanged between the service layer and the application layer can be understood. Table 8 lists the primitives supported by the INFO-SAP. Table 9 outlines primitive parameters.

Table 8 — INFO-SAP primitive summary

Name	Request	Indication	Response	Confirm
INFO-LEVELGET	6.4.1			6.4.2
INFO-LEVELSET	6.4.3	6.4.4		6.4.5
INFO-DATA	6.4.6	6.4.7		6.4.8

Table 9 — INFO-SAP primitive parameters

Name	Type	Valid range	Description

InfoSrcID	integer		Source node ID of information description service.
InfoDstID	integer		Destination node ID of information description service.
LevelVal	Enumeration	RAW_LEVEL, FEATURE_LEVEL, DECISION_LEVEL	Difference information description levels.
InfoLength	Unsigned Short integer	0-65535	Length of SDU.
InfoData	Variable length octet		Data portion of SDU.
InfoResultCode	Enumeration	SUCCESS, FAILED, LEVEL_NOT_SUPPORT	Result code of information description primitives.

6.4.1 INFO-LEVELGET.request

This primitive requests information description levels by CIP entities in the application layer.

6.4.1.1 Definition of service primitives

The syntax of this primitive is:

INFO-LEVELGET.request {

InfoSrcID,

InfoDstID,

}

Table 9 defines the parameters of this primitive.

6.4.1.2 When generated

This occurs by CIP entity from the application layer to query information description level of the InfoDstID node which is required by the InfoSrcID node.

6.4.1.3 Effect of receipt

On receipt of this primitive, the InfoDstID node gets current information description level.

6.4.2 INFO-LEVELGET.confirm

This primitive returns information description level to CIP entity in the application layer.

6.4.2.1 Definition of service primitives

The syntax of this primitive is:

INFO-LEVELGET.confirm {

InfoSrcID,

InfoDstID,

LevelVal,

InfoResultCode

}

Table 9 defines the parameters of this primitive.

6.4.2.2 When generated

This primitive returns the result of querying information description level.

6.4.2.3 Effect of receipt

If the InfoResultCode is SUCCESS, it means that the information description level of the InfoDstID node is successfully returned in the SYNTimeRef parameter. Otherwise, a FAILED error is confirmed.

6.4.3 INFO-LEVELSET.request

This primitive sets the information description level by CIP entity in the application layer.

6.4.3.1 Definition of service primitives

The syntax of this primitive is:

INFO-LEVELSET.request {

InfoSrcID,

InfoDstID,

LevelVal

}

Table 9 defines the parameters of this primitive.

6.4.3.2 When generated

This occurs by CIP entity from the application layer to set the information description level of the InfoDstID node to LevelVal.

6.4.3.3 Effect of receipt

On receipt of this primitive, the InfoDstID node sets the information description level. Information description levels conceptually correspond to phases of information processing. Different levels feature different types, structures and length of information. Three information description levels are defined: RAW_LEVEL, FEATURE_LEVEL and DECISION_LEVEL. Once new information description level is set successfully, correspondent information processing procedures or algorithms may be trigger or be applied. A node can simultaneously support more than one information description levels.

6.4.4 INFO-LEVELSET.indication

This primitive indicates the set operation of information description levels from the service layer to the local CIP entity.

6.4.4.1 Definition of service primitives

The syntax of this primitive is:

INFO-LEVELSET.indication {

InfoSrcID,

InfoDstID,

LevelVal

}

Table 9 defines the parameters of this primitive.

6.4.4.2 When generated

This occurs when the service layer indicates the set operation of information description levels to CIP entity.

6.4.4.3 Effect of receipt

On receipt of this primitive, the CIP entity is indicated the set operation of information description levels in the service layer.

6.4.5 INFO-LEVELSET.confirm

This primitive confirms the set of information description levels from the service layer to the CIP entity.

6.4.5.1 Definition of service primitives

The syntax of this primitive is:

INFO-LEVELSET.confirm {

InfoSrcID,

InfoDstID,

LevelVal,

InfoResultCode

}

Table 9 defines the parameters of this primitive.

6.4.5.2 When generated

This primitive reports the result of setting information description levels to the CIP entity.

6.4.5.3 Effect of receipt

If the InfoResultCode is SUCCESS, it means that the information description level in the InfoDstID node is successfully set to LevelVal. If the InfoResultCode is LEVEL_NOT_SUPPORT, it means that the LevelVal information description level is not supported or can not be provided by the InfoDstID node probably due to the capability of that node. If the InfoResultCode is FAILED, an error is indicated to the InfoSrcID node.

6.4.6 INFO-DATA.request

This primitive requests the transfer of information of specific information description levels by CIP entity in the application layer.

6.4.6.1 Definition of service primitives

The syntax of this primitive is:

INFO-DATA.request {

InfoSrcID,

InfoDstID,

LevelVal,

InfoLength,

InfoData

}

Table 9 defines the parameters of this primitive.

6.4.6.2 When generated

This occurs by CIP entity from the application layer to request the transfer of information of specific information description level of the InfoDstID node to the InfoSrcID node.

6.4.6.3 Effect of receipt

On receipt of this primitive, the entity of the service layer in the InfoSrcID node sends the InfoData to the peer entity of the service layer in the InfoDstID node.

6.4.7 INFO-DATA.indication

This primitive indicates to the CIP entity in the application layer that data unit of specific information description level has been received.

6.4.7.1 Definition of service primitives

The syntax of this primitive is:

INFO-DATA.indication {

InfoSrcID,

InfoDstID,

LevelVal

}

Table 9 defines the parameters of this primitive.

6.4.7.2 When generated

This occurs when the service layer indicates the reception of data unit of specific information description level to CIP entity.

6.4.7.3 Effect of receipt

On receipt of this primitive, the CIP entity is indicated that data unit of LevelVal information description level has been received.

6.4.8 INFO-DATA.confirm

This primitive confirms the transfer of data unit of specific information description level from the service layer to the CIP entity.

6.4.8.1 Definition of service primitives

The syntax of this primitive is:

INFO-DATA.confirm {

InfoSrcID,

InfoDstID,

LevelVal,

InfoResultCode

}

Table 9 defines the parameters of this primitive.

6.4.8.2 When generated

This primitive reports the result of transferring data unit of specific information description levels to the CIP entity.

6.4.8.3 Effect of receipt

If the InfoResultCode is SUCCESS, it means that a data unit of the LevelVal information description is successfully transferred from the InfoSrcID node to the InfoDstID node. Otherwise, an FAILED error is indicated to the InfoSrcID node.

6.5 Node-to-node inter-activation service

In this subclause, node-to-node inter-activation service in the service layer is defined. Node-to-node inter-activation is provided through N2NACT-SAP. The N2NACT-SAP is the logical interface between node-to-node inter-activation service entity in the service layer and CIP entity in the application layer. This logical interface incorporates a set of primitives and their definitions. These primitives and definitions are described

conceptually here, but through this the process of the parameters exchanged between the service layer and the application layer can be understood. Table 10 lists the primitives supported by the N2NACT-SAP. Table 11 outlines primitive parameters.

Table 10 — N2NACT-SAP primitive summary

Name	Request	Indication	Response	Confirm
N2NACT	6.5.1			6.5.2

Table 11 — N2NACT-SAP primitive parameters

Name	Type	Valid range	Description
N2NSrcID	integer		Source node ID of node-to-node inter-activation service.
N2NDstID	integer		Destination node ID of node-to-node inter-activation service.
N2NACTMode	Enumeration	NORMAL, RESERVATION	Difference node-to-node inter-activation modes.
N2NACTModeDataLength	Unsigned Short integer	0x00-0xFF	Data length for specific node-to-node inter-activation mode
N2NACTModeData	Variable length octet		Data for specific node-to-node inter-activation mode
N2NDataLength	Unsigned Short integer	0-65535	Context data length for node-to-node inter-activation process.
N2NData	Variable length octet		Context data for node-to-node inter-activation process.
N2NResultCode	Enumeration	SUCCESS, FAILED, MODE_NOT_SUPPORT	Result code of node-to-node inter-activation service primitives.

6.5.1 N2NACT.request

This primitive implements the inter-activation process between two nodes which can be requested by CIP entity in the application layer.

6.5.1.1 Definition of service primitives

The syntax of this primitive is:

N2NACT.request {

N2NSrcID,

N2NDstID,

N2NACTMode,

N2NACTModeDataLength,

N2NACTModeData,

N2NDataLength,

N2NData

}

Table 11 defines the parameters of this primitive.

6.5.1.2 When generated

This occurs by CIP entity from the application layer to request activation in the N2NDstID node from the N2NSrcID node.

6.5.1.3 Effect of receipt

On receipt of this primitive, the entity of the service layer in the N2NSrcID node sends the activation request to the peer entity of the service layer in the N2NDstID node. If the N2NDstID node receives the request successfully, the activation process is executed in the N2NACTMode mode. Two different modes are supported in this service. One is named as NORMAL activation mode, in which the N2NDstID node activates itself or configures its states using the context data (N2NData) included in this primitive. The other mode is RESERVATION mode, the N2NDstID node uses N2NACTModeData to reserve activation process. This process may be triggered automatically in a future time. The context data for activation process is given in the N2NData primitive parameter.

6.5.2 N2NACT.confirm

This primitive confirms the result of node-to-node inter-activation process from the service layer to the CIP entity.

6.5.2.1 Definition of service primitives

The syntax of this primitive is:

N2NACT.confirm {

N2NSrcID,

N2NDstID,

N2NACTMode,

N2NResultCode

}

Table 11 defines the parameters of this primitive.

6.5.2.2 When generated

This primitive reports the result of node-to-node inter-activation process to the CIP entity.

6.5.3.3 Effect of receipt

If the N2NResultCode is SUCCESS, it means that the N2NDstID node is successfully activated by the N2NSrcID node with the N2NACTMode mode. If N2NACTMode is NORMAL, then the N2NDstID node is already in active. If N2NACTMode is RESERVATION, the N2NDstID node will be activated automatically in a future time. If the N2NResultCode is MODE_NOT_SUPPORT, it means that the N2NDstID node

6.6 Parameter adaptation service

In this subclause, parameter adaptation service in the service layer is defined. Parameter adaptation service is provided through PAR-SAP. The PAR-SAP is the logical interface between parameter adaptation service entity in the service layer and CIP entity in the application layer. This logical interface incorporates a set of primitives and their definitions. These primitives and definitions are described conceptually here, but through this the process of the parameters exchanged between the service layer and the application layer can be understood. Table 12 lists the primitives supported by the PAR-SAP. Table 13 outlines primitive parameters.

Table 12 — PAR-SAP primitive summary

Name	Request	Indication	Response	Confirm
PAR	6.6.1	6.6.2		6.6.3

Table 13 — PAR-SAP primitive parameters

Name	Type	Valid range	Description
PARSrcID	integer		Source node ID of parameter adaptation service.
PARDstID	integer		Destination node ID of parameter adaptation service.
PARParameterName	string		Parameter name.
PARParameterLength	Unsigned Short integer	0-65535	Parameter length in parameter adaptation process.
PARParameter	Variable length		New parameter to be used in the adaptation

	octet		process.
PARResultCode	Enumeration	SUCCESS, FAILED, INVALID_PARAMETER	Result code of parameter adaptation service.

6.6.1 PAR.request

This primitive requests the parameter adaptation process by CIP entity in the application layer.

6.6.1.1 Definition of service primitives

The syntax of this primitive is:

PAR.request {

PARSrcID,

PARDstID,

PARParameterName,

PARParameterLength,

PARParameter

}

Table 13 defines the parameters of this primitive.

6.6.1.2 When generated

This occurs by CIP entity from the application layer to request the parameter adaptation in the PARDstID node.

6.6.1.3 Effect of receipt

On receipt of this primitive, the entity of the service layer in the PARDstID node tries to retrieve its local parameter with the PARParameterName name. If the node successfully finds the PARParameterName parameter, the value of this parameter is updated with PARParameter. Otherwise, an error code is generated.

6.6.2 PAR.indication

This primitive indicates to the CIP entity in the application layer that the request of parameter adaptation has been received.

6.6.2.1 Definition of service primitives

The syntax of this primitive is:

PAR.indication {

PARSrcID,

PARDstID,

PARParameterName

}

Table 13 defines the parameters of this primitive.

6.6.2.2 When generated

This occurs when the service layer indicates the parameter adaptation process to CIP entity.

6.6.2.3 Effect of receipt

On receipt of this primitive, the CIP entity is indicated that the request of PARParameterName parameter adaptation has been received.

6.6.3 PAR.confirm

This primitive confirms the result of parameter adaptation from the service layer to the CIP entity.

6.6.3.1 Definition of service primitives

The syntax of this primitive is:

PAR.confirm {

PARSrcID,

PARDstID,

PARParameterName,

PARResultCode

}

Table 13 defines the parameters of this primitive.

6.6.3.2 When generated

This primitive reports the result of parameter adaptation to the CIP entity.

6.6.3.3 Effect of receipt

If the PARResultCode is SUCCESS, it means that the PARParameterName parameter in the PARDstID node is successfully updated. If the PARResultCode is INVALID_PARAMETER, it means that the PARParameterName parameter is not defined or valid in the PARDstID node. Otherwise, an FAILED error is indicated to the InfoSrcID node.

7 Enhanced Services and Interfaces Specifications

This clause specifies enhanced services supporting CIP in intelligent sensor networks. Service primitives and parameters of primitives are defined for each core service. In Table X, the names of service access points (SAPs) through which specific service is provided are listed.

Table X — Core services and the names of SAPs

Service name	SAP name
QoS management service	QoS-SAP
CIP-driven node state scheduling service	NSTATE-SAP
Context management service	CONTEXT-SAP

7.1 QoS management service

(to be added)

7.1.1 Service specification

(to be added)

7.1.2 Primitives and parameters

The next process is to group data elements including to find objects and to group data elements by object, and then to find properties involved in the objects and sub-grouped data elements by properties.

(to be added)

7.2 CIP-driven node state scheduling service

(to be added)

7.2.1 Service specification

(to be added)

7.2.2 Primitives and parameters

(to be added)

7.3 Context management service

(to be added)

7.3.1 Service specification

(to be added)

7.3.2 Primitives and parameters

(to be added)

Annex – A Core Services and Interfaces Examples

This annex is informative. Examples of core services and interfaces in an intelligent sensor network based anti-intrusion system are given.

(to be added)

(Other examples can be considered. Contributions are requested.)

Annex – B Enhanced Services and Interfaces Examples

This annex is informative. Examples of enhanced services and interfaces in an intelligent sensor network based anti-intrusion system are given.

(to be added)

(Other examples can be considered. Contributions are requested.)

Annex – C Bibliography

(to be added)