Navigations- und Funkkommunikationsgeräte und -systeme für die Seeschifffahrt

Digitale Schnittstellen

Teil 401: Mehrere Datensender und mehrere Datenempfänger
Schiffssystemzusammenschaltung – Anwendungsprofil
(IEC 61162-401:2001) Deutsche Fassung EN 61162-401:2002, Text in Englisch

DIN

EN 61162-401

ICS 47.020.70

Maritime navigation and radiocommunication equipment and systems –
Digital interfaces – Part 401: Multiple talkers and multiple listeners –
Ship systems interconnection – Application profile
(IEC 61162-401:2001); German version EN 61162-401:2002, text in English

Matériels et systèmes de navigation et de radiocommunication maritimes –
Interfaces numériques – Partie 401: Emetteurs multiples et récepteurs multiples –
Interconnexion des systèmes embarqués – Couche application
(CEI 61162-401:2001); Version allemande EN 61162-401:2002, texte en anglais

Die Europäische Norm EN 61162-401:2002 hat den Status einer Deutschen Norm.

Beginn der Gültigkeit

Die EN 61162-401 wurde am 2002-02-01 angenommen.

Nationales Vorwort

Für die vorliegende Norm ist das nationale Arbeitsgremium K 738 „Elektronische Navigationsinstrumente“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE zuständig.

Das Präsidium des DIN hat mit Präsidialbeschluss 1/78 festgelegt, dass Deutsche Normen, deren Inhalt sich auf internationale Arbeitsergebnisse der IEC gründet, unter bestimmten Bedingungen allein in englischer Sprache veröffentlicht werden dürfen. Die Bedingungen sind für diese Norm erfüllt. Sie steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen. Es kann davon ausgegangen werden, dass die Anwender über hinreichende englische Sprachkenntnisse verfügen. Für viele der verwendeten Benennungen existieren keine gebräuchlichen deutschen Benennungen, da sich die deutschen Anwender im Bereich der Seeschifffahrt in der Regel ebenfalls der englischen Benennungen bedienen.

Norm-Inhalt für die Teile 400, 401, 410 und 420 der Reihe DIN EN 61162-4 war veröffentlicht als E DIN IEC 80/175/CD:1998-10 und E DIN IEC 80/176/CD:1998-10.

Die enthaltene IEC-Publikation wurde vom TC 80 „Maritime navigation and radiocommunication equipment and systems“ erarbeitet.

Die Reihe DIN EN 61162 besteht aus 4 Teilen, wobei der Teil 4 in mehrere Einzelnormen unterteilt ist. Eine zusammenfassende Beschreibung der Normen dieser Reihe und eine Übersicht über deren unterschiedliche Anwendbarkeit ist in Anhang B der Norm DIN EN 61162-400 enthalten.

Das IEC-Komitee hat entschieden, dass der Inhalt dieser Publikation bis Juni 2005 unverändert bleiben soll. Zu diesem Zeitpunkt wird entsprechend der Entscheidung des Komitees die Publikation
– bestätigt,
– zurückgezogen,
– durch eine Folgeausgabe ersetzt oder
– geändert.

Fortsetzung Seite 2
und 117 Seiten EN

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE
Normenstelle Schiffs- und Meerestechnik (NSMT) im DIN Deutsches Institut für Normung e. V.

AVA Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

Nationaler Anhang NA
(informativ)

Zusammenhang mit Europäischen und Internationalen Normen

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

IEC hat 1997 die Benummerung der IEC-Publikationen geändert. Zu den bisher verwendeten Normnummern wird jeweils 60000 addiert. So ist zum Beispiel aus IEC 68 nun IEC 60068 geworden.

Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

EUROPÄISCHE NORM

EUROPEAN STANDARD

NORME EUROPÉENNE

EN 61162-401

Februar 2002

ICS 47.020.70

Deutsche Fassung

Navigations- und Funkkommunikationsgeräte und -systeme für die Seeschifffahrt

Digitale Schnittstellen

Teil 401: Mehrere Datensender und mehrere Datenempfänger – Schiffssystemzusammenschaltung – Anwendungsprofil
(IEC 61162-401:2001)

Maritime navigation and radiocommunication equipment and systems – Digital interfaces – Part 401: Multiple talkers and multiple listeners – Ship systems interconnection – Application profile
(IEC 61162-401:2001)

Matériels et systèmes de navigation et de radiocommunication maritimes – Interfaces numériques – Partie 401: Emetteurs multiples et récepteurs multiples – Interconnexion des systèmes embarqués – Couche application
(CEI 61162-401:2001)

Diese Europäische Norm wurde von CENELEC am 2002-02-01 angenommen. Die CENELEC-Mitglieder sind gehalten, die CEN/CENELEC-Geschäftsordnung zu erfüllen, in der die Bedingungen festgelegt sind, unter denen dieser Europäischen Norm ohne jede Änderung der Status einer nationalen Norm zu geben ist.

Auf dem letzten Stand befindliche Listen dieser nationalen Normen mit ihren bibliographischen Angaben sind beim Zentralsekretariat oder bei jedem CENELEC-Mitglied auf Anfrage erhältlich.

Diese Europäische Norm besteht in drei offiziellen Fassungen (Deutsch, Englisch, Französisch). Eine Fassung in einer anderen Sprache, die von einem CENELEC-Mitglied in eigener Verantwortung durch Übersetzung in seine Landessprache gemacht und dem Zentralsekretariat mitgeteilt worden ist, hat den gleichen Status wie die offiziellen Fassungen.

CENELEC-Mitglieder sind die nationalen elektrotechnischen Komitees von Belgien, Dänemark, Deutschland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Luxemburg, Malta, Niederlande, Norwegen, Österreich, Portugal, Schweden, Schweiz, Spanien, der Tschechischen Republik und dem Vereinigten Königreich.

CENELEC

Europäisches Komitee für Elektrotechnische Normung
European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique

Zentralsekretariat: rue de Stassart 35, B-1050 Brüssel

© 2002 CENELEC – Alle Rechte der Verwertung, gleich in welcher Form und in welchem Verfahren, sind weltweit den Mitgliedern von CENELEC vorbehalten.

Ref. Nr. EN 61162-401:2002 D

AVA Documents Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

Vorwort

Der Text des Schriftstücks 80/310/FDIS, zukünftige 1. Ausgabe von IEC 61162-401, ausgearbeitet von dem IEC TC 80 „Maritime navigation and radiocommunication equipment and systems“, wurde der IEC-CENELEC Parallelen Abstimmung unterworfen und von CENELEC am 2002-02-01 als EN 61162-401 angenommen.

Nachstehende Daten wurden festgelegt:

– spätestes Datum, zu dem die EN auf nationaler Ebene durch Veröffentlichung einer identischen nationalen Norm oder durch Anerkennung übernommen werden muss (dop): 2002-11-01

– spätestes Datum, zu dem nationale Normen, die der EN entgegenstehen, zurückgezogen werden müssen (dow): 2005-02-01

Anhänge, die als „normativ" bezeichnet sind, gehören zum Norminhalt.
Anhänge, die als „informativ" bezeichnet sind, enthalten nur Informationen.
In dieser Norm sind die Anhänge A, B, C und ZA normativ und Anhang D ist informativ.
Der Anhang ZA wurde von CENELEC hinzugefügt.

Anerkennungsnotiz

Der Text der Internationalen Norm IEC 61162-401:2001 wurde von CENELEC ohne irgendeine Abänderung als Europäische Norm angenommen.

Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

CONTENTS

INTRODUCTION 8

1 Scope 9
1.1 General 9
1.2 Application profile 9
2 Normative references 9
3 Definitions 10
3.1 Terms 10
3.2 Abbreviations 13
3.3 Conventions 13
3.4 Protocol data types 15
3.5 Non-protocol types 17
3.6 Literals 18
4 Dependence on T-profile 18
4.1 General 18
4.2 LNA-LNA or MAU-LNA communication 18
4.3 Services required 18
4.4 Services required, but not used by the A-profile 19
4.5 Variable quality of service attributes 19
4.5.1 Reliability and safety 19
4.5.2 Real time response 19
4.5.3 Support for stream data 20
4.5.4 Maximum and minimum message lengths 20
4.6 Congestion and flow control 20
5 Functional requirements for MAU 20
5.1 General overview 20
5.2 MAU configuration management 21
5.2.1 Define MAU parameters (`MauInitialize`) 22
5.2.2 Connection open request (`MauOpen`) 23
5.2.3 Connection close (`MauClose`) 23
5.2.4 MAU state change call-back (`MauState`) 24
5.3 Session management, authentication and congestion control 25
5.3.1 Session control and authentication 25
5.3.2 Congestion control (`MauSession`) 26
5.3.3 Request and connection limiting 26
5.4 Interface and connection point overview 26
5.4.1 Special considerations for anonymous broadcast (ABC) 27
5.4.2 MCP membership in interfaces 27
5.4.3 MCP attributes 27
5.4.4 Modify MCP attributes (`McpGet, McpSet`) 28
5.4.5 Interface attributes 29
5.4.6 Modify interface attributes (`IfGet, IfSet`) 29

Documents
Document Sharing Hub
Full document details: DIN EN 61162-401:2002-11

5.5 Interface management 30
5.5.1 Define interface (IfTable) 31
5.5.2 Remove interface (IfRemove) 31
5.5.3 Establish interface connection (IfOpen) 32
5.5.4 Close interface (IfClose) 32
5.5.5 Interface state change call-back (InterfaceState) 33
5.6 MCP transactions. 34
5.6.1 Function type transactions 35
5.6.2 Non-acknowledged write transactions 36
5.6.3 Subscribe type transactions 36
5.6.4 Individual subscribe transactions 37
5.6.5 Client side initiation (TransActivate) 38
5.6.6 Transaction cancellation (TransCancel). 38
5.6.7 Client side transaction state change (TransClientState) 39
5.6.8 Server side transaction state change (TransServerState) 39
5.7 Bulk data transfer 40
6 Functional requirements for LNAs 40
6.1 Context diagram and functional overview 40
6.2 MAU name management (MauAck, MauRequest, SessionClose) 41
6.2.1 Duplicate MAU names 43
6.2.2 Local MAU names 43
6.2.3 Remote MAU names 43
6.3 LNA-LNA session management 45
6.3.1 Heard about LNA 45
6.3.2 Known LNA 45
6.3.3 Dead LNA 46
6.3.4 The LNA watchdog function 46
6.4 Local MAU connection management 47
6.4.1 MAU identification 47
6.4.2 Local MAU states 47
6.4.3 Starting MAU management 48
6.4.4 Ending MAU management 48
6.5 MAU control MCP 49
6.5.1 General overview 49
6.5.2 Functionality 49
6.5.3 Watchdog service 49
6.6 Session management 50
6.6.1 General principles 50
6.6.2 MAU session information 50
6.6.3 Session codes used for authentication 50
6.7 Accept type interface management 51
6.7.1 General overview 51
6.7.2 Possible errors in interface definitions or connection requests 51
6.7.3 Special handling of anonymous broadcast interfaces 52
6.7.4 Definition and removal of accept interface 53
6.7.5 Establishing and closing connections to accept interfaces 53
6.8 Connect type interface management 55

Documents
Document Sharing Hub
Full document details: DIN EN 61162-401:2002-11

6.9 General transaction management 57
6.9.1 MCP identity and transaction address 57
6.9.2 Transaction identity 57
6.9.3 Session identity 57
6.9.4 Transaction types 57
6.9.5 Exception handling 58
6.10 Accept side transaction management 59
6.10.1 Read, write and function transaction handling 61
6.10.2 Non-acknowledged write request 61
6.10.3 Initial subscribe 61
6.10.4 Initial individual subscribe 62
6.10.5 Server initiated subscribe acknowledgement 62
6.10.6 Server initiated individual subscribe acknowledgement 62
6.10.7 Anonymous broadcast subscribe 62
6.10.8 Cancellation of a transaction 62
6.11 Connect side transaction management 63
6.12 LNA-MAU 64
6.13 Use of priority levels in LNA 64
6.14 Congestion control 65
6.14.1 Congestion cases 65
6.14.2 LNA requirements 65
6.14.3 MAU requirements 66
7 Protocol defined as sequence diagrams 66
7.1 General conventions 66
7.1.1 Broken connections 66
7.1.2 Exception handling 66
7.2 LNA management 67
7.2.1 Opening LNA-LNA connection 67
7.2.2 LNA watchdog functionality 67
7.2.3 Congestion control 67
7.3 Opening and closing MAU sessions 68
7.3.1 MAU to MAU communication via same LNA 68
7.3.2 Opening a MAU session 68
7.3.3 Closing a MAU session 69
7.3.4 Finding a remote MAU in LNA 70
7.3.5 Server MAU or LNA dies 71
7.4 Opening and closing interfaces 71
7.4.1 Opening accept interface 71
7.4.2 Closing accept IF 72
7.4.3 Opening connect IF 72
7.4.4 Closing connect interface 73
7.5 Data transfer messages 73
7.5.1 Data transfer of read, write or function type 74
7.5.2 Subscribe data transfers 75
7.5.3 Data transfer cancel and transfer timeout 76

Documents
Document Sharing Hub
Full document details: DIN EN 61162-401:2002-11

8 Message definitions 78
8.1 Introduction 78
8.1.1 Common message format 78
8.2 MAU-LNA messages 79
8.2.1 Message format 79
8.2.2 MAU session control 80
8.2.3 Interface definition messages 83
8.2.4 Data transfer messages 86
8.3 LNA-LNA message formats for reliable link 89
8.3.1 General message format 89
8.3.2 Connection management 90
8.3.3 MAU management 90
8.3.4 Interface connection management 92
8.3.5 Data transfer management 94
8.4 LNA-LNA message formats for multicast link 95
8.4.1 General message format 95
8.4.2 Different multi-cast ports 96
8.4.3 Name look-up and watchdog messages 96
8.4.4 Data transfer messages 98
8.4.5 Anonymous broadcast message details 99
9 General identity codes 100
9.1 Protocol and software version codes 100
9.2 Network address, node number and LNA id (`address_m`, `word32_m`) 100
9.3 MAU identity (`mauname_m`, `mauid_m`) 101
9.4 Data object and MCP identity (`mcid_m`) 101
9.5 Interface code (`mcid_m`) 102
9.6 Data object name (`mcname_m`, `mciname_m`). 102
9.7 Interface name (`ifname_m`) 102
9.8 Session identity (word16_m) 102
9.9 Transaction identity (`word32_m`) 102
9.10 Format string 103
9.10.1 Data record format encoding 103
9.10.2 Transaction type coding 104
9.10.3 Complete format string 104
9.11 Password (`password_m`) 104
10 Data marshalling 105
10.1 Introduction 105
10.2 Network octet order 105
10.2.1 General 105
10.2.2 Basic types 105
10.2.3 Composite types 106
10.2.4 Messages 106
10.3 Pack and unpack routines 106
10.3.1 Introduction 106
10.3.2 Pack routine 107
10.3.3 Unpack routine 107

Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

11 Communication link between MAU and LNA 107
11.1 Introduction 107
11.1.1 General service specification 108
11.1.2 Point to point 108
11.1.3 Connection oriented 108
11.1.4 Symmetrical and full duplex 108
11.1.5 Message based 108
11.1.6 Priority 108
11.1.7 Buffering and flow control 108
11.1.8 Reliable transfer 108
11.1.9 Error reporting 109
12 General principles for module functionality 109
12.1 Flexibility in receiving, conservatism in sending 109
12.2 Garbled messages 109
12.3 Closed communication links 109

Annex A (normative) Message codes 110
Annex B (normative) Error codes and message field values 112
Annex C (normative) Symbolic constants 113
Annex D (informative) Compatibility between MiTS and the IEC 61162-400 series 115
Anhang ZA (normativ) Normative Verweisungen auf internationale Publikationen mit ihren entsprechenden europäischen Publikationen 118

Documents
Document Sharing Hub
Full document details: DIN EN 61162-401:2002-11

INTRODUCTION

International Standard IEC 61162 is a four-part standard which specifies four digital interfaces for applications in marine navigation, radiocommunication and system integration.

The four parts are:

IEC 61162-1 Single talker and multiple listeners

IEC 61162-2 Single talker and multiple listeners, high speed transmission

IEC 61162-3 Multiple talkers and multiple listeners – Serial data instrument network

IEC 61162-4 Multiple talkers and multiple listeners – Ship systems interconnection.

Part 4 of the standard is sub-divided into a number of individual standards with part numbers in the IEC 61162-400 series. A full reference to part 4 can be found in IEC 61162-400, clause 4.

This part of the standard, IEC 61162-401: A-profile specification, defines the application functionality and its implementation in an application layer protocol.

Relationship with the other parts of the IEC 61162 series of standards is defined in annex B to IEC 61162-400.

Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

MARITIME NAVIGATION AND RADIOCOMMUNICATION EQUIPMENT AND SYSTEMS – DIGITAL INTERFACES –

Part 401: Multiple talkers and multiple listeners – Ship systems interconnection – Application profile

1 Scope

1.1 General

IEC 61162-4 series specifies a communication system for use in integrated ship control systems.

IEC 61162-400 defines the overall functional scope for the communication system.

1.2 Application profile

This part of IEC 61162 describes the application profile (A-profile – corresponding to ISO-OSI layers 5 to 7 [ISO 7498]) of the communication protocol which is the basis for the communication system. It relies on the realization of layers 1 to 4 (the T-profile) as described in part 410.

The description of the A-profile is in terms of services offered to the application using the protocol and of message contents and sequences used to realize these services.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of IEC 61162. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of IEC 61162 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of IEC and ISO maintain registers of currently valid International Standards.

IEC 61162-400, *Maritime navigation and radiocommunication equipment and systems – Digital interfaces – Part 400: Multiple talkers and multiple listeners – Ship systems interconnection – Introduction and general principles*

IEC 61162-410, *Maritime navigation and radiocommunication equipment and systems – Digital interfaces – Part 410: Multiple talkers and multiple listeners – Ship systems interconnection – Transport profile requirements and basic transport profile*

IEC 61162-420, *Maritime navigation and radiocommunication equipment and systems – Digital interfaces – Part 420: Multiple talkers and multiple listeners – Ship systems interconnection – Companion standard requirements and basic companion standards*

IEEE 754: *IEEE Standard for Binary Floating-Point Arithmetic*

ISO/IEC 8859-1, *Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*

Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

ISO/IEC 10646-1, *Information Technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*

RFC 2500:1999, *Internet Official Protocol Standards – Internet Activities Board standard*

NOTE RFC (request for comments) is a document issued by the Internet engineering task force (IETF), the International standardization body for the Internet, that describes a part of the Internet protocol. Some RFCs are accepted as official Internet standards and listed in the "Internet Official Protocol Standards" itself an RFC.

3 Definitions

This clause is divided into definition of terms (terms), definition of abbreviations (abbreviations), definitions of nomenclature (conventions), definition of data types (data types) and definition of literal formats. Other definitions valid for this part of IEC 61162 are contained in part 400 of this standard.

3.1 Terms

For the purpose of this part of IEC 61162, the following terms apply:

3.1.1
anonymous broadcast (ABC)
a broadcast service where the sender does not know to which MAU it is sending data. Similarly the listener may not know which sender it should listen for

3.1.2
array
a linear indexed sequence of identical data types. The index runs from zero and upwards. Arrays can have variable lengths (with a fixed upper limit) or fixed lengths. The difference between these two types is normally only visible during transmission between modules where the real length of a variable length array is transmitted as an attribute

3.1.3
bit order
this standard numbers bits in an octet from zero to seven. Bit seven is the most significant bit; bit zero the least significant

3.1.4
character
this standard provides two mechanisms for the transmission of characters:

a) an 8-bit character based on ISO/IEC 8859-1 (also called ISO Latin-1). This set covers most national alphabets based on the Latin letters;

b) a 16-bit character based on ISO/IEC 10646-1. This standard specifies the use of the 16-bit form Universal Character Set 2 (UCS-2) which covers most of the commonly used character sets in the world

NOTE 1 Later revisions of the standard may also support 32-bit characters.

NOTE 2 Any reference to *character* in this standard implies the 8-bit character if not otherwise stated.

3.1.5
companion standard
these are the mechanisms to define and describe how the A-profile services are used to implement some application functions and interfaces (see IEC 61162-420)

3.1.6
connection
an association between two interfaces or two MCPs, one each on a server and a client MAU. A connection must be established before transactions can be activated

AVA Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

There is also a connection between each pair of LNAs and between each MAU and its LNA. Although similar in concept, they are not directly associated with the activation of transactions.

3.1.7
data type
this standard defines a set of data types that have a machine and T-profile independent interpretation. The types cover, for example characters, signed and unsigned integers and floating point containers as well as some other derived types. The data types are defined in 3.4

3.1.8
format string
a text string that defines the structure of certain data records that are transmitted via the T-profile. It also contains information about the function supported by data objects. See clause 8.4.3.5 for more information

3.1.9
interface
a collection of MCPs (references to data objects) implemented on a server or client MAU which behaves similarly to MCPs in that operations on interfaces effect all MCPs in the interface

3.1.10
IP – internet protocol
references in this standard refer to the protocol defined in RFC 2500. Part 410 of IEC 61162 specifies a T-profile using version 4 of the internet protocol (IPV4), but additional parts may in the future define T-profiles using newer versions of the protocol

3.1.11
IPV4 – internet protocol version 4
is currently the most used version of the internet protocol (IP). See IEC 61162-410 for more details

3.1.12
magic number
The first field of all messages which will be common to a group of messages. It can be used by the protocol software to identify and verify messages and message boundaries. See annex A

3.1.13
maximum message size
this protocol is based on the exchange of messages between various modules. The T-profile may impose limits on the maximum size of these messages. See clause 4 for more information

3.1.14
message
one of the basic components of this standard. They are collections of information items with a defined format that are exchanged between various modules to achieve some service or part service. The message formats are defined in clause 8. The exchange patterns are defined as sequence diagrams in clause 7

3.1.15
module
entities (programs or host computers) between messages that are sent which are either an LNA (application independent network communication manager) or a MAU (the application program module)

Documents
Document Sharing Hub
Full document details: DIN EN 61162-401:2002-11

3.1.16
network bit order
the A-profile definition is resolved to the octet level. The ordering of bits within each octet is defined by the T-profile in use. See clause 4 for more information

3.1.17
network octet order
all multi-octet entities (messages or fragments of messages) are defined to be transferred in a T-profile and host computer independent format. This format is defined in 8.4.3.5

3.1.18
null MCP
an MCP associated with the MAU itself. It will have a reference code of zero. It can be manipulated similarly to a normal MCP through the MAU control data object (see 6.5) and it can also be used to transfer information to MAU call back routines (see 5.2.4). The null MCP cannot be connected to from remote MAUs and it is not associated with any interface

3.1.19
octet
the smallest information entity that the A-profile protocol considers which consists of 8 bits transmitted by the T-profile in some specific sequence. This standard requires that a group of octets can be transmitted as a message and that the received message has the same sequence of octets as the sent message and that each received octet has the same bit pattern as the sent octet. The protocol does not specify any particular ordering of bits within octets as long as it is consistent over the network

NOTE One entity of the Boolean type can be represented as a bit within an octet, but any number of Boolean entities will be transmitted as an integral number of octets.

3.1.20
record
a sequence of different or identical data types given a fixed definition and an associated type name

3.1.21
sequence diagram
a representation of how messages are transferred between the modules of the protocol system. The format is defined in 3.3.3

3.1.22
session
the term is used to identify a connection from a client MAU to a server MAU. It represents one unbroken connection where the LNAs guarantee that no other client MAU has replaced the originally connecting MAU. The death of a client MAU will be reported to the server MAU as a closed session

3.1.23
T-profile network
a collection of network nodes (host computers) that can communicate with each other following this standard. The concept is defined in IEC 61162-410.

This standard does not specify how nodes on two different T-profile networks shall communicate

NOTE The standard does, however, specify that one node shall be able to be connected to more than one T-profile network at one time and this can be used to develop gateway nodes.

3.1.24
TCP/IP – transmission control protocol / internet protocol
for version 4 of the internet protocol (IPV4) is defined in RFC793. See IEC 61162-410 for more details

Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

3.1.25
transaction
the term is used to identify an exchange of data between two MAUs that is related to one initial request from the client. It can consist of from zero (non-acknowledged write) to an unlimited number (subscriptions) of messages from the server

3.1.26
version number
in this standard, it is 4.0 (major number is 4, minor is 0). Earlier version codes have been used by MiTS (see IEC 61162-400)

3.2 Abbreviations

Table 1 lists important abbreviations used in this standard. The third column gives a reference to the clause where a definition of the term can be found.

Table 1 – Abbreviations

Term	Description	Reference
ABC	Anonymous broadcast	3.1.1
Ack	Acknowledge	Figures & text
IPC	Inter-process communication	11.1
IP	Internet protocol	3.1.10
IPV4	Internet protocol version 4	3.1.11
LNA	Local network administrator	Part 400
MAU	MiTS application unit (application program module)	Part 400
MCP	MAU connection point	5.3
MTU	Maximum transmission unit	4.1
Obj	Object	Figures & text
Req	Request	Figures & text
RLNA	Remote LNA (in sequence diagram)	3.3.3
RMAU	Remote MAU (in sequence diagram)	3.3.3
TCP/IP	Transmission control protocol/internet protocol	3.1.24
UCS	Universal character set	3.1.4

3.3 Conventions

3.3.1
use of type face
each description element has its own conventions for use of type face. In this standard, the following conventions apply:

a) use of `courier` type face indicates either some defined protocol data type (see next item) or another protocol identifier (e.g. a message code, see 3.3.2);

b) use of the extension `_m` in a word type set in `courier` means that it references one of the defined protocol data types. The types are defined in 3.4;

c) *italics* are normally used when a word or term references a label or identifier in an associated table or figure.

3.3.2
message definition template
this standard describes a message based protocol. Each message will be described as an ordered list of fields. An example is shown in table 2.

Documents
Document Sharing Hub
Full document details: DIN EN 61162-401:2002-11

Table 2 – Example of message definition

Octet #	Data type	Field description
0 – 1	`word16_m`	`MV4_MAGIC`
2 – 3	`word16_m`	*type*, message type
4 – 5	`word16_m`	Priority
6 – 7	`word16_m`	*length*, message length (= x+1 or y+1)
8 – x		message body necessary
– y		message body sometimes required

The message definition has three columns.

a) The first column gives the octet numbers used by a field, numbered from the first to the last octet inclusive of the field. The numbering starts with 0 at the first octet of the message. The letters x, y, etc. are used to indicate the last byte number in a variable length field. Arithmetic expressions may also be indicated in this column. Arithmetic expressions are enclosed in parenthesis.

b) The second column gives the data type of the field (see 3.1.7 and 3.4.2).

c) The third column describes the contents of the field. Identifiers in `courier` show constant values. The value of these can be found in the annexes. Expressions typeset in *italics* are described in the text following the message definition.

The physical message consists of the entities defined in the table transmitted as a sequence of (multi-octet) quantities. A message may consist of one part that is always transmitted and one part that is sometimes transmitted. The limit between these two parts is shown by a thick row divider line in the table

3.3.3
sequence diagrams
message sequences are described in diagrams. An example is shown in figure 1

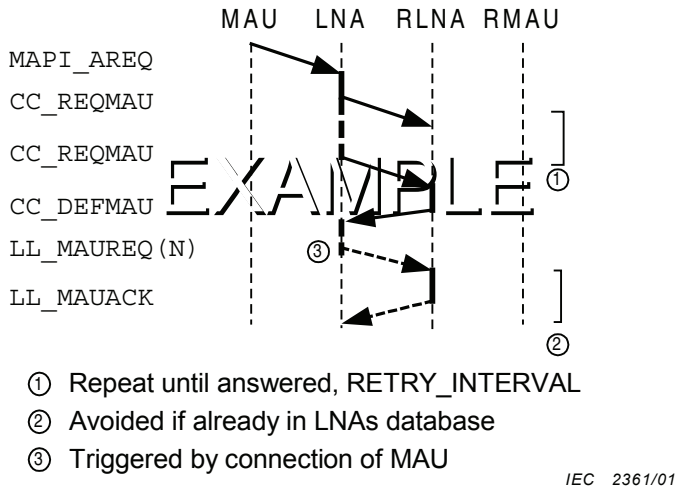

Figure 1 – Sequence diagram example

The following graphical elements are used:

– each **communicating module** is represented by a vertical dashed line. The upper list of names identify the modules. The same module list will be used in all diagrams (the modularization is defined in IEC 61162-400):

Documents
Document Sharing Hub
Full document details: DIN EN 61162-401:2002-11

1) **MAU** is the local application unit communicating with its LNA;
2) **LNA** is the local communication manager communicating with MAU and RLNA;
3) **RLNA** is the remote communication manager communicating with LNA and RMAU;
4) **RMAU** is the remote application unit communicating with its RLNA;

NOTE When two MAUs which are connected to the same LNA are communicating, the LNA – RLNA communication sequence may depend on the implementation. However, it is recommended that the implementation uses the standard message sequence and defines some kind of LNA internal loop-back interface for internal messages.

– **time** runs from top to bottom. If needed, timing information may be shown explicitly on the right side as comments, otherwise no time scale is given;
– **messages** are shown as arrows between the vertical lines. The arrows are pointing from the sender to the receiver, and downwards. Dashed arrows (as in comment (2)) indicates messages that are sent only under certain conditions;
– **message code** is shown on the far left side, aligned with the arrow start point. The code is associated with a message structure that is defined in clause 8. Two special short-hand notations are used:
  1) a trailing (N) in parenthesis defines a message with a negative acknowledgement,
  2) a trailing (C) in parenthesis specifies two alternate messages. Explanation is given in the text;
– **a thick solid vertical line** along the module line is used if the reception of a message causes an “immediate” sending of one or more new messages. This thick line should be interpreted as showing that the connected operations are executed in the same control thread and within a relative short time frame;
– **a thick dashed vertical line** is used if the new message is sent after a (possibly) significant time, for example after a timer expires (as in comment (1)) or as a result of another internal event (as in comment (3)),
– **sequence indicators** to group one or more messages or other graphical elements (usually as a line on the right hand side) associated with a comment reference. This means that the comment applies to the group of graphical elements (as example for comment (1) and (2));
– **comment references** as numbers enclosed in circles. The number will refer to a comment either in the drawing itself or in the following text, with the number in parenthesis.

The graphical elements are supplemented by other textual explanation where necessary.

3.4 Protocol data types

3.4.1
general

this clause defines the data types used by this standard. Data type is defined as a data container which is capable of containing information with a certain precision. Any implementation of this protocol shall make sure that the host computer’s or the T-profile’s representation of these data types have at least the precision defined in the following clauses

3.4.2
basic type

for numerical and textual values, the standard makes references to the set of basic types described in Table 3. The standard will use the names listed under the *type* column to identify any of the basic types. The format column specifies the format string encoding for the relevant type and the short format column lists the one-character code that can be used to identify the type, for example in software libraries (see 8.4.3.5).

Documents
Document Sharing Hub
Full document details: DIN EN 61162-401:2002-11

Table 3 – Basic entities

Type	Format	Short format	Description
bool_m	b	l	1 bit Boolean (0 = FALSE, 1 = TRUE)
char8_m	c8	c	8 bit character [ISO/IEC 8859-1]
char16_m	c16	C	16 bit character [ISO/IEC 10646-1], UCS-2
word8_m	w8	b	8 bit unsigned integer
word16_m	w16	w	16 bit unsigned integer
word32_m	w32	W	32 bit unsigned integer
word64_m	w64	U	64 bit unsigned integer
int8_m	i8	t	8 bit 2's complement integer
int16_m	i16	I	16 bit 2's complement integer
int32_m	i32	I	32 bit 2's complement integer
int64_m	i64	H	64 bit 2's complement integer
float32_m	f32	f	32 bit floating point [IEEE 754]
float64_m	f64	F	64 bit floating point [IEEE 754]

NOTE 1 The char8_m data-type can be both signed and unsigned. This is dependent on the implementation. It shall, however, be guaranteed true to a relevant machine's representation of an 8-bit ISO/IEC 8859-1 character.

NOTE 2 Expect an extension of this list to include higher precision entities, for example 32-bit characters, 128-bit integers and 128-bit floating point. It is recommended that coding practices take this into consideration.

3.4.3
array

arrays are sequences of basic types, unions or records (not arrays). Arrays are variable length or fixed length, with a special type of variable length array called an opaque block:

a) **arrays**: this is an indexed list of element types. It is defined by element type and array length. Elements are numbered from zero and up. The length is constant and all elements are transmitted on the network;

b) **variable length arrays** (var-array): this is an array with a variable length. It is defined by element type, maximum length and the type of an unsigned integer used to contain the length of the transmitted array. Only the length field itself and the number of elements specified in the length field are transmitted on the network.

c) **opaque block:** this is a special case of the variable length array that is used to support more complex formats that cannot be represented in the current formatting strategy. An opaque block is a variable length array of unsigned octets (word8_m) that must be further interpreted by a specified information definition.

In this part of IEC 61162, only fixed length arrays are used. They are indicated by putting the array length in square brackets after the array type name, for example word16_m[32].

A special case of the array is the null terminated string. This is a sequence of characters terminated by the null character. The maximum length of the string is always defined. This type is only used in the A-profile definition and is not accessible to the application programmer

Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

3.4.4
record
is an ordered list of basic types, other records or arrays. All elements or arrays are transmitted. It is defined as an ordered list of elements and/or arrays

Records are not used in this part. They are only used by application programmers. Messages can be regarded as a special case of records. Messages are defined in clause 8.

3.4.5
union
is similar to a record, but with the definition of an additional unsigned integer used to index the element or array that is actually transmitted in one given transfer. Only one element or array in the basic record can be transmitted. The union is defined as a record, but with the additional index type

Unions are not used in the A-profile definition. They are available for application programmers

3.4.6
derived types
this part of the standard also specifies a set of derived types that are used to contain certain special protocol codes. These are described in Table 4. The format column specifies the format string encoding for the relevant type (see 8.4.3.5). Note that the `Ns` format represents the special array type called “null terminated string” (in this text references to as `string_m`)

Table 4 – Derived number formats

Type	Format	Description	Reference
address_m	(w16[w8:48]w8)	Network address for a T-profile network	9.2
nodenumber_m	w32	Network node number in a T-profile network	9.2
mauid_m	w16	LNA specific identifier for a local MAU	9.3
mauname_m	32s	MAU name	9.3
mcid_m	w16	Data object identifier code	9.4
mciname_m	32s	Data object name for interface MCP	9.6
ifname_m	32s	Interface name	9.7
password_m	8s	Password, fixed length array of 8 char8_m	9.11
word16_m	w16	Identifier for remote MAU-LNA session	9.8
string_m	Ns	Null terminated string of maximum length N (inc. null)	0
word32_m	w32	Identifier for a transaction	9.9

3.5 Non-protocol types

Some data types are used in this standard to refer to various data entities that need to be implemented in the API or in the LNA, but which do not occur in protocol messages. The table below gives an overview of these.

Table 5 – Non-protocol data types

Type	Description	Reference
val_m	A general integral value, either integer or reference to structure	5.4.3
eventh_m	A reference or reference to a call-back function	5.4.3
pack_m	A reference to a data marshalling function	5.4.3
priority_m	Priority (PT_LOW, PT_NORMAL, PT_URGENT)	5.4.3

The data types are defined based on semantics of the C language and may be more or less difficult to map to other languages

Documents
Document Sharing Hub
Full document details: DIN EN 61162-401:2002-11

3.6 Literals

This standard uses literals as defined in annex A of IEC 61162-400. The principles are summarized below.

3.6.1
numbers
decimal and hexadecimal (prefixed with `0x`) numbers are used. Decimal numbers can be signed and/or floating point

3.6.2
characters and strings
single characters are enclosed in single quotes, strings in double

3.6.3
Boolean
the Boolean values `TRUE` (numeric value 1) and `FALSE` (numeric value 0) are used

3.6.4
symbolic constants
annexes A to C define symbolic constants used in this standard. All references to these constants in this standard are by the constant's symbolic name in `courier` typeface

4 Dependence on T-profile

4.1 General

This part of the standard is written in a form that makes it independent of the T-profile. The principle is that different T-profiles can be used, dependent on the application area and corresponding requirements. The standard principles for the definition and construction of a T-profile are contained in IEC 61162-410.

This clause gives an overview of the assumptions that this part has been based upon and some of the deviations that can be allowed.

4.2 LNA-LNA or MAU-LNA communication

The LNA is only required to use the T-profile for LNA-LNA communication. However, it is suggested that the LNA also uses the same logical structure (TP-network, network node number) for MAU-LNA communication (see 6.4.1.2).

4.3 Services required

The T-profile is assumed to provide node connection and data transport services as described in IEC 61162-410 of this standard. Briefly summarized, these services are:

a) define and establish one or more T-profile networks where nodes are individually identifiable by network node number;

b) establish reliable (and possibly redundant) communication links for messages between any two nodes on the network;

c) establish listening and transmitting multi-cast ports that can send or receive messages between and to all nodes on a T-profile network;

Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

d) detect and report errors on a connection, including loss of redundancy;

e) send and receive reliable messages on two priority levels (PT_URGENT and PT_NORMAL);

f) send and receive unreliable broadcast (or multicast) messages on the PT_URGENT and PT_NORMAL priority level.

This standard enables communication between nodes on one T-profile network.

4.4 Services required, but not used by the A-profile

The T-profile shall in addition to the services listed in the previous subclause provide two further management-oriented services:

a) physical network management;

b) time distribution.

These services are required for the integrated control system, but will not be used to implement A-profile services as described in this part of the standard. These services are assumed to be handled independently of the implementation of the A-profile in the host computer.

4.5 Variable quality of service attributes

The A-profile does not necessarily need all services described above, nor does it need to have the same quality of service for all variants of T-profiles. This subclause describes allowable deviations from the assumed quality of service and the consequences for A-profile functionality.

4.5.1 Reliability and safety

The T-profile is in principle assumed to give reliable delivery of certain types of data, also in face of partial loss of communication capability. Furthermore, it is assumed that no single error in the lower level transport functionality shall disrupt the delivery of data. This normally requires some form of redundancy, retransmission capabilities and sufficient self-checking of T-profile transmission mechanisms.

The A-profile will function without this level of reliability. However, the basic safety related properties of the communication system will be severely affected by lower service quality in these areas. For non-safety-related integration tests or non-critical data acquisition systems, it will normally be sufficient to have a lower integrity T-profile service, for example by using the worldwide Internet. For safety related systems, this will not be sufficient.

It is necessary to consider the application area for the system before a decision is made to allow a lower quality T-profile to be used. This must also be clearly documented.

For broadcast messages, the A-profile and the applications are required to assume that messages can be lost or duplicated. Both losses and duplication may impact on the performance of the system. The T-profile should reduce the possibilities for any of this to happen.

4.5.2 Real time response

The T-profile is required to provide three priority levels: PT_URGENT, PT_NORMAL and PT_LOW. Higher priority traffic shall have precedence over lower priority traffic. In addition, it is assumed that the T-profile can supply some kind of response guarantees for each of the priority levels.

Documents
Document Sharing Hub
Full document details: DIN EN 61162-401:2002-11

The three priority levels are sufficient to give some level of priority support even on simple T-profiles without quality of service support on the transport level, for example as for Internet version 4 over Ethernet. The use of individual buffers on the data link level and prioritized scheduling of messages from the buffers should give sufficient priority control for most purposes. However, actual parameters must be determined based on implementation characteristics and load figures.

4.5.3 Support for stream data

Stream data interfaces are not required by the A-profile specification. To use stream data interfaces, the MAUs have to establish a direct stream between them. The establishment of such streams is defined in the companion standards. The implementation of the stream is defined by the T-profile.

4.5.4 Maximum and minimum message lengths

This standard does not enforce any upper limit on message sizes. However, the underlying T-profile may define some limits that will be propagated to the A-profile, for example:

a) for urgent messages, the message length will normally be limited to the maximum transmission unit (MTU) of the transport protocol in use;

b) for some transport protocols there may be general message length restrictions due to protocol particulars, for example for message-oriented transfers;

c) some implementations may limit the message length to increase efficiency.

The maximum message length, if enforced by the implementation, shall be readable by the application through specified API calls (see service definitions in IEC 61162-410).

The A-profile will be designed to accommodate message sizes as low as 512 bytes of payload data. This constant is defined as `MV4_MES_MIN` in annex C. Note that this will reduce the available payload also for application programs.

4.6 Congestion and flow control

The A-profile will use flow control mechanisms in the T-profile to inhibit further transmissions on a communication link if it is congested. The flow control shall only be used on low and normal priority links. The flow control will be applied on the session level (see 5.2.4). A session can be looked at as an association between two MAUs.

In case of congestion, the A-profile shall queue non-transmitted messages until the T-profile is able to receive new outgoing messages. The implementation shall as far as possible avoid that messages are cancelled due to congestion. However, an application error that makes the MAU continue to send messages on a congested link may be handled by closing the session completely.

5 Functional requirements for MAU

5.1 General overview

The context diagram for the MAU is shown in Figure 2 The MAU uses its local LNA for all communication with the network. It may be equipped with a diagnostics console, but the functionality of this is not defined in this standard.

Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

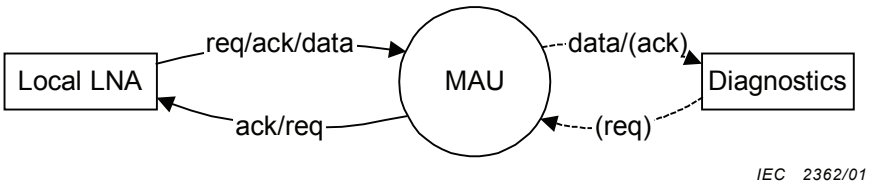

Figure 2 – MAU context diagram

The MAU is divided into the application specific part and the application programmer's interface (API). The latter part implements the application independent part of the MAU and takes care of mapping the application services to and from the relevant LNA-MAU messages. The implementation of the application services in the LNA are discussed in clause 6.

This clause defines the services that this protocol makes available to the application programs (MAU). Most of these services are implemented as a message to the LNA, possibly with some form of acknowledgement being expected later. In this case, the message name is included in the service definition. The protocol can then be determined by looking up the corresponding sequence diagram.

Other services require the use of an application program interface (API) with more functionality than simply a translation from subroutine call to protocol message. These services are not referenced to a particular message.

All services are required and must be implemented by all open APIs supporting this protocol. Clause A.7 of IEC 61162-400 gives an overview of the service definition format and how to convert it into an API.

5.2 MAU configuration management

The services defined in this subclause are used to manipulate MAU state, from connection to LNA to removal. The services are based on the state transition diagram shown in Figure 3.

The states are:

- **`MAU_DEFINED`**: a call to the service `MauInitialize` defines the MAU's context and defines it as a local non-connected application unit. This state can be used to manipulate attributes of connection points or define connection points, but no connections can be established;
- **`MAU_OPENING`**: a connection attempt to the LNA is initiated by a call to `MauOpen`. The MAU will stay in this state until the connection is established, it times out or an unrecoverable network error occurs. The application program must wait for one of these events or explicitly request a close (`MauClose`),
- **`MAU_OPEN`**: when the connection to the LNA is established, the MAU enters the open state. At this stage, the MAU is defined in the network and has been assigned a set of standard services accessible by the LNA and other MAUs in the system. Connection points can now be established, and when successfully opened, used for transactions,
- **`MAU_CLOSED`**: the connection to the LNA has been requested to close by an external unit (LNA or another MAU) or by the application itself calling `MauClose`. The application program waits for the `EV_LNA_RESTART` signal before it reopens the MAU-LNA connection;

NOTE The external request can usually be denied by the MAU, in which case the MAU stays in the open state. The kill signal cannot be denied.

- **`MAU_ERROR`**: a non-recoverable communication error occurred. New communication parameters must be established before a new attempt can be made.

Documents
Document Sharing Hub
Full document details: DIN EN 61162-401:2002-11

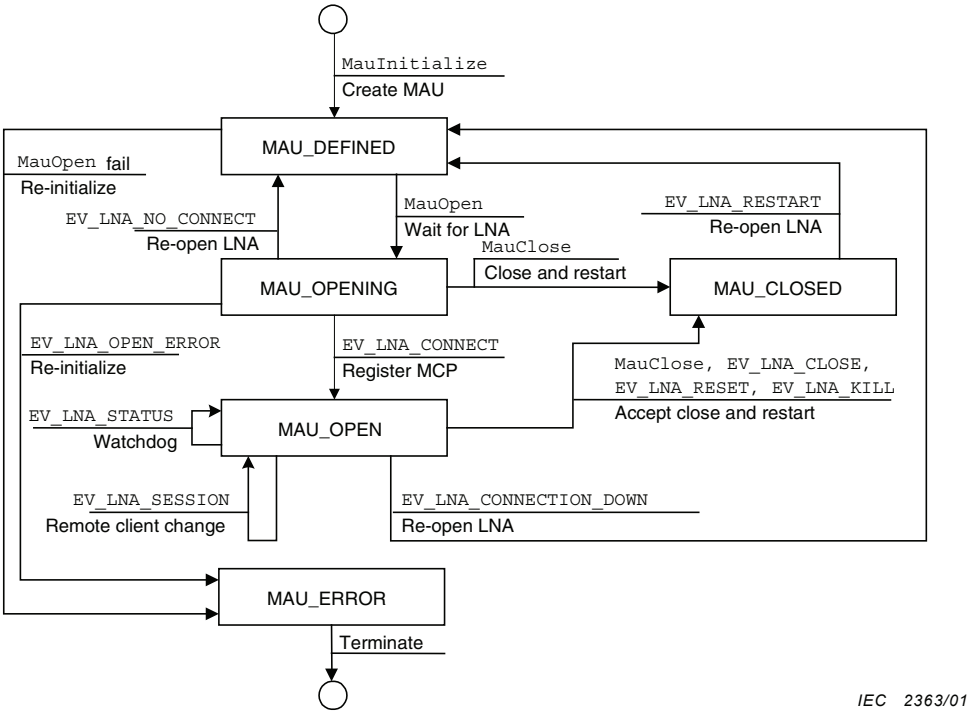

Figure 3 – MAU state diagram

Note that the MAU has an associated MCP (the null MCP, see 6.5) that can be used to control and monitor some of the connection related attributes of the MAU. The *last error* attribute for the null MCP will return any error details associated with a MAU level message exchange.

5.2.1 Define MAU parameters (`MauInitialize`)

NOTE This call is used to define certain parameters that are used when the MAU is later instantiated.

Pre-condition: OS and communication sub-system initialized. A communication descriptor for MAU-LNA communication has been created.

Post-condition: system ready to open a communication channel to the LNA or an unrecoverable error occurred.

Input parameters:

a) communication descriptor: a reference to the communication channel used between MAU and its LNA, usually provided through a T-profile API (TLI);

b) maximum number of data objects: this is used to allocate memory for descriptor tables. May be unnecessary in some implementations;

c) MAU name: logical name of application module. It is possibly to let the LNA define a unique name by using a default MAU name;

d) MAU description: a descriptive string,

e) MAU password: used to protect against unauthorized access to MAU control functions;

f) a call-back routine for state changes as shown in the figure,

g) an optional up-call routine (possibly with other semantics than a clean call-back) which is used to report internal errors and warnings from the API.

Documents
Document Sharing Hub
Full document details: DIN EN 61162-401:2002-11

Output parameters: none.

Call-back: none.

Exceptions: none.

Errors: errors will be related to various illegal parameter values or system level errors.

5.2.2 Connection open request (`MauOpen`)

This call is used to request the opening of the MAU-LNA connection and to send identity information to the LNA.

Pre-condition: MAU parameters defined (`MAU_DEFINED`).

Post-condition: waiting for LNA (`MAU_OPENING`) or error state.

Input parameters: optional timeout for connection attempt.

Output parameters: none.

Call-back: state changes shall be reported as described in the diagram. This includes fatally bad communication parameters.

Exceptions: possible internal resource problems.

Errors: the MAU is in an illegal state. Communication parameters are bad.

This service uses the `MAPI_OREQ` and `MAPI_OACK` messages. The result from the acknowledgement is delivered through the MAU state call-back.

5.2.3 Connection close (`MauClose`)

This service requests the close of the LNA-MAU session. This is done by closing the connection to the LNA and cleaning up internal state. The mechanism for closing the connection is dependent on the T-profile in use.

The closing down will take some time and the MAU will remain in the closed state until the process is completed. Note that all pending transactions and all MCP connections will be terminated without individual warning. The API is required to clean up the internal state, but the application must also do that with any state it maintains.

Pre-condition: MAU is or in the process of being connected to LNA.

Post-condition: MAU disconnected from LNA (`MAU_CLOSED`).

Input parameters: none

Output parameters: none.

Call-back: state change will be reported to `MauState`.

Documents
Document Sharing Hub
Full document details: DIN EN 61162-401:2002-11

Exceptions: none

Errors: the MAU is in an unexpected state.

This service may use the session close message (`MAPI_SESSION`), but will normally just close the communication link to the LNA.

5.2.4 MAU state change call-back (`MauState`)

This call-back routine shall be called each time the MAU changes state. It is also called when the LNA requests a state change for the MAU or a watchdog return message.

a) `EV_LNA_CLOSE`: request close, can be denied by a non-zero acknowledgement.

b) `EV_LNA_RESET`: request restart (close and then open), can be denied by a non-zero acknowledgement.

c) `EV_LNA_KILL`: force close (possibly not delivered, but detected as a connection close: `EV_LNA_CONNECTION_DOWN`).

d) `EV_LNA_STATUS`: check status, used by watchdog, The API is required to close the LNA connection when the application gives a non-zero acknowledgement.

NOTE A negatively acknowledged status request will be handled as a kill, i.e. force a transition to close. A negatively acknowledged close or reset request will be handled as a status request, i.e. make the MAU remains open.

The possible events are summarized in the table below. The *return value* column specifies the return value that is expected for continuous normal operation (remaining in the open state).

Table 6 – MAU events

Event code	Normal return value	Description
`EV_LNA_CLOSE`	<> 0	Close request
`EV_LNA_CONNECT`	n/a	Connect state change
`EV_LNA_CONNECTION_DOWN`	n/a	Communication link down, state change
`EV_LNA_KILL`	n/a	Kill command
`EV_LNA_NO_CONNECT`	n/a	Connection failed due to timeout or missing LNA
`EV_LNA_OPEN_ERROR`	n/a	Port description parameter error
`EV_LNA_RESET`	<> 0	Reset request
`EV_LNA_RESTART`	n/a	Ready to re-connect state change
`EV_LNA_SESSION`	n/a	Remote client MAU session changes state
`EV_LNA_STATUS`	0	Status/watchdog request

The `EV_LNA_SESSION` event is passed to the MAU when a remote MAU, which previously has established a connection to one of this MAU's interfaces, dies or changes state. The session code can be retrieved by reading MCP zero's session (`M_SESSION`) attribute value. The state change information can be got from the last error attribute. The following session change codes are used:

a) `SESSION_C0`: the communication link has been restored to normal (no congestion);

b) `SESSION_C1`: the communication link associated with the session is congested and no more low priority messages will be accepted for transmission;

c) `SESSION_C2`: the communication link is also closed for normal priority messages;

d) `SESSION_DOWN`: the communication link to the remote MAU has been or should be closed.

Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

NOTE 1 The session down code should normally be used to clean up internal state associated with the closed session (pending transactions mainly, but also session related structures used for example, client authentication). This may be done by calling the interface connection handlers with a “remote client dead” event. This may be handled automatically by the API.

NOTE 2 If the MAU continues to send messages on a communication link indicated as congested, this can be treated as an application error. The LNA may then close all links to the MAU and terminate the session. On the other hand, the LNA must be able to accept some more messages that may be somewhere in the communication pipeline without raising this exception.

Pre-condition: `MauInitialize` has successfully been called to define the call-back routine. The new state has normally been entered when the event is delivered (except for close requests, where denial – negative return – makes the MAU stay in the open state.

Post-condition: see above.

Input parameters: new state code and corresponding event code as defined in the figure.

Output parameters: not applicable, zero or non-zero, dependent on event.

Call-back: not applicable.

Exceptions: not applicable.

Errors: not applicable.

5.3 Session management, authentication and congestion control

A session is an association between the local MAU and one remote MAU. This means that most MCP or interface activities (see next subclause) are also related to a session (the exception is the anonymous broadcast MCPs). The local MAU cannot manipulate a session state, but the state of remote MAUs or LNAs define the state of the session.

Sessions are used for two purposes in this standard:

a) for authentication of sender of messages (connections or transactions);

b) for congestion control between two MAUs or between LNA and MAU.

5.3.1 Session control and authentication

The session states are defined by the following events:

a) the creation of the first connection between an MCP or interface on this MAU and another MAU: This defines the session. No other session-related event is associated with this;

b) any congestion related messages (`EV_LNA_SESSION`: low or normal priority congestion or no congestion) changes the “capacity” of the communication link defined by the session. The local MAU shall adjust its transmission according to these messages;

c) the loss of the connection to the remote MAU causes a session down message (`EV_LNA_SESSION`) to be delivered to the MAU. This message means that the MAU must clean up all states related to the session (close interfaces or MCPs and terminate all pending transactions). This event signals the removal of the session.

The API may decide whether the session shall be handled as an object in its own right, possibly with application related services, for example to interrogate the state. Note however, the required clean-up after a session close. Session management is handled through the `MAPI_SESSION` message and through the *session* field in the general LNA-MAU message. It is suggested that the API keeps track of the session identifier and makes this available to the application program (e.g. through the M_SESSION MCP attribute)

Documents
Document Sharing Hub
Full document details: DIN EN 61162-401:2002-11

The LNA guarantees that the session identifier is unique to a session and that it is updated whenever an association changes (also if the remote MAU is disconnected and then reconnected).

5.3.2 Congestion control (`MauSession`)

The MAU may issue a congestion notification, either to a specific remote MAU or to the LNA (session zero). This notification should make the relevant entity stop the sending of low or normal priority messages. Congestion control does not work for urgent messages.

The MAU cannot rely completely on the congestion control notification and it should be used mostly between MAUs that have a mutual understanding or between the MAU and the LNA.

The MAU must itself cancel the congestion notification.

Pre-condition: MAU connected, and one or more sessions established.

Post-condition: session marked as congested (or unmarked) by remote MAU and/or LNA.

Input parameters: session to be manipulated, level (congested or normal)

Output parameters: none.

Call-back: none.

Exceptions: none.

Errors: bad input parameters.

This service uses the `MAPI_SESSION` message.

5.3.3 Request and connection limiting

An alternative way to limit possible congestion is to limit the number of clients or outstanding transactions on the interfaces. Interfaces can be marked with a maximum number of clients and pending transactions. These are attribute codes defined with the interface (Table 8). By setting these attributes, the MAU can limit the number of connections and/or transaction requests. The mechanism does not work for urgent requests. The limits apply to all MCPs in an interface.

5.4 Interface and connection point overview

A data object is the abstract representation in the network of connection points that are defined and served by a MAU. In both server and client MAU, these data objects are referenced by a MAU connection point (**MCP**). An MCP in an open state (with a connection between the server and one or more clients) is necessary to receive or send messages between MAUs.

The system supports two types of connection points as indicated in Figure 4:

a) the control connection point (Null MCP) that is used by the LNA for status requests, the watchdog function and MAU state control. This object is created automatically when the MAU is opened and is normally only visible to the application programmer through the close, reset, kill and status MAU events. This MCP is established automatically by the API and LNA. Transactions are handled by the API and transferred to the MAU call-back routine;

Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

b) application defined interface MCP which is associated with an application level interface. They can only use the short data object name format, but will additionally have an interface name. These MCPs are defined in groups, one group for each interface. They are connected and disconnected in the same groups. Transactions are handled individually.

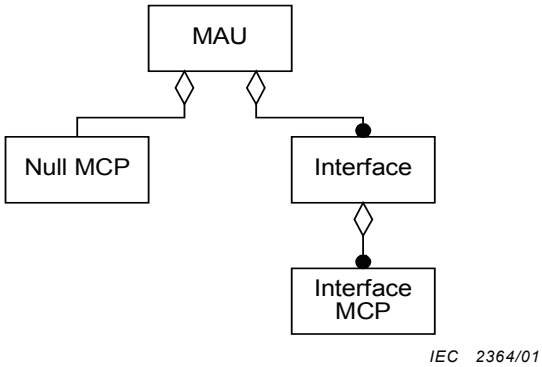

Figure 4 – Types of connection points

The control MCP does not usually need to be considered by the application programmer. The clauses following will discuss the services available for the interface MCP. The remaining part of this subclause will give an overview of MCP attributes.

5.4.1 Special considerations for anonymous broadcast (ABC)

Anonymous broadcasts are not associated to special MAUs. They are messages that can be sent to or received from any MAU. To facilitate the processing of anonymous broadcasts, the following restrictions apply:

a) an anonymous broadcast MCP can only be part of an interface called “ABCn”, where ‘n’ is replaced with one of the numbers ‘1’ to ‘5’. The interface name will determine the broadcast port used for the transmission and reception of messages. The MAU name is ignored;

b) the significant number of characters in the MCP name is 8 (the left-most characters). More characters can be used, but they will be ignored;

c) the message length is limited to 512 octets maximum;

d) the format string is hashed into a 32-bit number.

These restrictions mean (particularly item d)) that there is a possibility that differing data objects can be mixed at the receiving side. However, the probability should be low and the design tools should make sure that all ABC entries have different MCP names.

5.4.2 MCP membership in interfaces

One interface MCP will normally be associated one-to-one with an interface. However, it is possible to define “logical” MCPs that are members of two or more interfaces. This is done by creating one MCP that is registered as belonging to the different interfaces. This is useful for general MCPs that can be used in two or more different contexts. If the same MCP is used in two interfaces, it will be assigned two different MCP identity codes, one for each interface. The LNA will treat the two identity codes as distinct MCPs, but the application can use the same attribute values, including call-back functions and storage for both.

5.4.3 MCP attributes

This subclause defines all MCP attributes. The attributes are summarized in the table below.

Documents
Document Sharing Hub
Full document details: DIN EN 61162-401:2002-11

Table 7 – MCP attributes

Attribute	M	Type	Default	Description
M_MCPNAME	F	mciname_m	*	The symbolic data object name
M_FORMAT	F	char8_m*	*	The format string
M_SEND_STORE		void*	NULL	Output data-buffer
M_RECV_STORE		void*	NULL	Input data-buffer
M_SEND_PACK		pack_m	NULL	Pack-function (marshalling for send)
M_RECV_PACK		pack_m	NULL	Unpack-function (marshalling for receive)
M_PRIORITY		int	normal	Default priority for transactions
M_TRANS_HANDLER		eventh_m	NULL	Transaction-handler
M_TRANS_RCODE		val_m	0	User return-code for transaction handler
M_LASTERROR	R	int32_m		Last error code
M_SESSION	R	word16_m		Session ID for current operation
M_INTERFACE	R	val_m		Return interface reference or NULL
M_CAN_CONFIGURE		bool_m	FALSE	TRUE if remote configurable allowed

The columns from left to right are:

– the attribute column specifies a symbolic identifier for the attribute in question;

– the ‘M’ column specifies how the attribute can be modified; an ‘R’ in this column means that the attribute is set by external events, for example a communication error, and an ‘F’ that the attribute cannot be changed after a connection attempt succeeded. No letter in this column means that the attribute can be changed at any time;

– the type column specifies the type of the attribute. The types are defined in 3.4;

– the default column specifies the default value the attribute gets if it is not explicitly defined. An asterisk (‘*’) in this column means that the attribute must be specified;

– the description column gives a short description of the attribute.

The null MCP shall support these attributes, but the interpretation of some of the values may depend on context.

Note that the remotely configurable flag is only used by special companion standards for remote configuration. It is included here for completeness and need not be implemented in all APIs.

5.4.4 Modify MCP attributes (`McpGet, McpSet`)

These two services are used to read and modify MCP attributes. This service can be called at any time and also when the MCP is open; although only attributes that are marked as connect modifiable in Table 7 can be changed when the MCP is open or opening.

Note that different variants of these routines may be necessary to modify attributes of different types, for example it may be necessary to specify a special set and get routine for function references.

NOTE For some implementations of the API, it may be convenient to code some of the attribute retrieval or modification routines as specific services associated with MCPs or interfaces. This may be particularly appropriate for object-oriented programming languages where some of the attributes can be available as object attributes or modifiable through specific methods.

Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

Pre-condition: MCP is created (`MCP_CREATED`).

Post-condition: no change.

Input parameters: data object attribute code and optionally new value as listed in Table 7. Only those parameters listed with 'M' can be changed when the MCP is open.

Output parameters: old attribute values can be returned from the get service.

Call-back: none.

Exceptions: none.

Errors: unknown MCP or attribute code, illegal MCP state.

5.4.5 Interface attributes

This subclause defines all interface attributes. The attributes are listed in the table below.

Table 8 – Interface attributes

Attribute	S	M	Type	Default	Description
M_MAUNAME	A	F	mauname_m	NULL	The symbolic server MAU name
M_IFNAME		F	ifname_m	*	The symbolic interface name
M_IFCNAME		F	ifname_m	NULL	The name of the class of the interface
M_DESC		F	char8_m*	NULL	A description string for the interface
M_IS_ACCEPT		F	bool_m	FALSE	TRUE if accept type
M_PASSWORD		F	password_m	NULL	The interface password
M_CONN_HANDLER			eventh_m	*	Connection handler
M_CONN_RCODE			val_m	0	User return-code for connection handler
M_TIMEOUT	C	F	int32_m	0	Connection timeout (ms)
M_TRANSQUEUE	A	F	word16_m	0	Length of transaction queue
M_CLIENTS	A	F	word16_m	0	Number of clients
M_LASTERROR		R	int32_m		Last error code
M_SESSION		R	word16_m		Session ID for current operation
M_CAN_CONFIGURE			bool_m	FALSE	TRUE if remotely configurable

See 5.4.3 for a description of the columns and codes. The S column specifies if the attribute only has an interpretation for connect ('C') or accept ('A') side interfaces. For interface MCP, it is necessary to interrogate the interface attributes.

Note that the remotely configurable flag is used only by special companion standards for remote configuration. It is included here for completeness and need not be implemented in all APIs.

5.4.6 Modify interface attributes (`IfGet, IfSet`)

These services are used to read and/or modify interface attributes. This service can be called at any time, also when the interface is open.

See 5.4.4 for more information on function and possible implementations.

Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

Pre-condition: interface is created (`IF_CREATED`).

Post-condition: no change.

Input parameters: data object attributes as listed in Table 8. Only those parameters listed with "M" can be changed when the interface is open or opening.

Output parameters: attribute values can be returned from the get service.

Call-back: none.

Exceptions: none.

Errors: illegal attribute codes or values, MCP in wrong state.

5.5 Interface management

The state diagram for an interface is shown below.

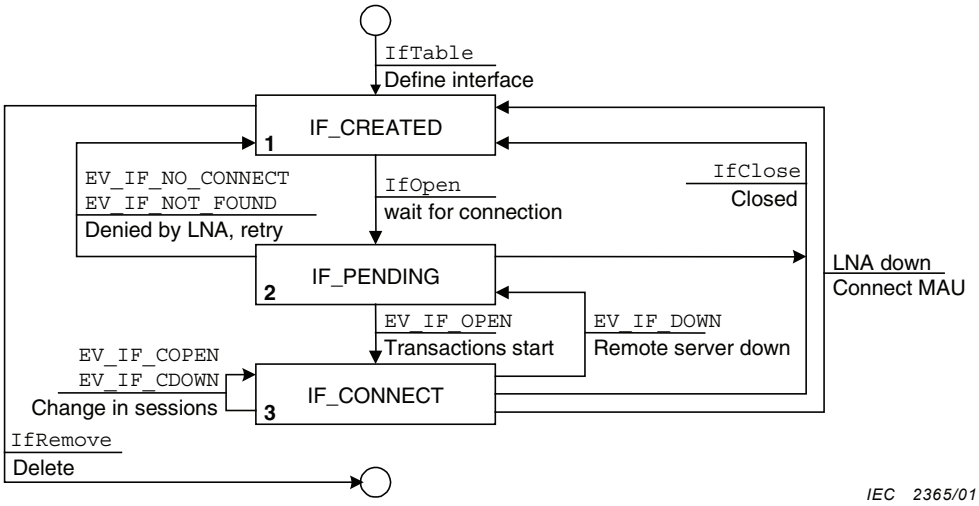

Figure 5 – Interface state transitions

The states are:

a) `IF_CREATED`: the interface has been defined by the application, but is not yet connected. The definition can be done while the MAU is in the created state (without connection to the LNA). Note also that if the LNA connection goes down, all interfaces will automatically be set to this state;

b) `IF_PENDING`: the interface attributes have been sent to the LNA. The MAU is waiting for acceptance. There is a difference here between accept and connect objects:

   1) accept object attributes are only exported to the local LNA. The object will only stay in this state until the local LNA accepts or denies the registration. This will normally only happen if there is an identical interface or MCP already defined by this MAU. A connection timeout has limited utility,

   2) connect interface attributes are exported to the system for connection to a remote MAU's accept interface. The interface will stay in this state until the remote MAU is found and has accepted or denied the connection or until the timeout triggers;

c) `IF_CONNECT`: the interface has been registered and it is possible to receive or initiate transactions.

Documents
Document Sharing Hub
Full document details: DIN EN 61162-401:2002-11

States b) and c) are only used when the MAU is in the open state. The closing of the MAU will transfer all interfaces back to created. The API is required to do necessary state clean up and terminate all pending transactions. The application is required to do a similar clean-up in its data structures.

This standard requires that all MCPs in a requested interface are accepted on the server side for the connection attempt to be positively acknowledged. This means that the client side interface must be a true subset of the accept side interface. Errors of any type shall be reported to the application.

If the MAU enters the error state, no call-backs will be delivered, neither on pending transactions nor closed connections. The interfaces will be in an undefined state.

The reconnection of a broken interface connection (from state c) to b) and back to c)) will be done automatically by the LNA. Note that the timeout, if specified, will bring the interface to state a), also after an automatic LNA reconnect.

The following subclauses define the services that are available for interface management.

5.5.1 Define interface (`IfTable`)

This service is used to define MCPs and MCP attributes for later export to the LNA. It defines a new interface and sets it to the created state. It can in principle be called at any time (see "M" marked attributes in clause 5.4.5), but it is most useful for initial creation. It specifies a set of attributes for the newly created data object.

Pre-condition: MAU is created (`MAU_DEFINED`).

Post-condition: `IF_CREATED`.

Input parameters: the attributes are partly common attributes to all MCPs in the interface and partly attributes associated with each MCP. It is suggested that the input format is in the form of a table that lists interface attributes, MCP attributes and attribute values in a format that can be automatically generated by a computer tool.

Legal interface attributes and attribute value types are those listed in Table 8. MCP attributes are listed in Table 7. The MCP attributes marked with 'F' cannot be set in an interface definition. Attributes in both tables marked with an asterisk must be defined before the interface can be opened.

Output parameters: interface reference.

Call-back: none.

Exceptions: none.

Errors: illegal MAU state, illegal interface state, illegal attributes or attribute values.

5.5.2 Remove interface (`IfRemove`)

This service is used to remove an interface, its MCPs and all associated state. It is only legal to do this when the interface is in the created state.

Pre-condition: `IF_CREATED`.

Post-condition: interface deleted.

Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

Input parameters: interface reference.

Output parameters: none.

Call-back: none.

Exceptions: none.

Errors: illegal interface reference or state.

5.5.3 Establish interface connection (`IfOpen`)

This service exports the definition of an interface to the LNA. It shall send the relevant messages to the LNA and wait for its reply. Two cases must be considered:

a) an accept type interface will cause the creation of network data objects and is handled by the local LNA alone. There are few restrictions on how the interface and the MCPs are constructed;

b) a connect type interface shall be connected to a set of network data objects and needs to match the intended targets in all connection oriented attributes (names and format strings). For the connection attempt to succeed the following pre-requisites exist:

   1) the connection related interface attributes must match on both sides,

   2) the connection related attributes in all the client defined MCPs must match (a subset of) the server defined MCPs. The client does not need to connect to the whole server interface, but all MCPs in the client interface must connect to a corresponding server MCP for the connection to succeed.

Pre-condition: interface is created (`IF_CREATED`) and all necessary attributes are defined.

Post-condition: `IF_PENDING` and then possibly `IF_CONNECT`

Input parameters: all connection-related attributes from the definition of the interface and its MCPs are used in executing this service. A time-out different from zero will make the LNA give up its connection attempt when the specified time has expired. Zero will make the LNA keep on trying to reconnect. The time-out has no function for accept type interfaces.

Output parameters: none.

Call-back: the interface connection handler is called at all further interface state changes. Transaction handlers are defined for each MCP. These will be called on incoming transactions.

Exceptions: timeout and format errors. The LNA may give other interface related error messages on some occasions. All exceptions will be delivered to the connection handler.

Errors: illegal interface state, illegal or undefined attribute values, illegal interface reference.

This service uses the `MAPI_IREQ` and `MAPI_IACK` messages. The result from the acknowledgement is passed back through the interface connection call back routine.

5.5.4 Close interface (`IfClose`)

This service removes an interface from the connected state. It can be used in the pending or the open state. The service will block any further transactions on this object. This service will signal to all remotely connected MAUs (server or clients) that the interface is down.

Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

Pre-condition: interface is pending or open (`IF_PENDING, IF_CONNECT`).

Post-condition: `IF_CREATED`.

Input parameters: reference to the interface.

Output parameters: none.

Call-back: none.

Exceptions: none.

Errors: illegal interface state, illegal interface reference.

This service uses the `MAPI_IREM` message.

5.5.5 Interface state change call-back (`InterfaceState`)

All state changes for the interface that are not an immediate effect of a service call will be reported to the interface connection call-back routine. In addition, it will be called when a remote MAU tries to connect to the interface.

Pre-condition: interface is pending or open (`IF_PENDING, IF_CONNECT`).

Post-condition: state has changed immediately before call.

Input parameters: event code and interface reference, user defined reference (value of attribute `M_CONN_RCODE`). All event codes are listed in the table below.

Table 9 – Interface events

Event code	Return value	Description
`EV_IF_OPEN`	n/a	Interface open
`EV_IF_DOWN`	n/a	Interface closed from remote side
`EV_IF_COPEN`	0	Client requests connection to accept interface
`EV_IF_CDOWN`	n/a	Client disconnects from accept interface
`EV_IF_NO_CONNECT`	n/a	Attribute values inhibit the establishment of the interface
`EV_IF_NOT_FOUND`	n/a	Connection timed out or was denied due to, for example high load

NOTE The `EV_IF_CDOWN` event is only delivered (as an A-profile message) when the remote client closes down a connection in a controlled manner. If the remote MAU dies, this will be notified through a session close event to the MAU (see below).

The *no connect* code is used when attribute values inhibit the connection attempt, for example password errors, badly formatted names, etc. The *not found* code is used when the attempt was aborted because the data object was not found (timeout) or because connection was temporarily denied. Details about the cause can be by examining the *last error* attribute value for the interface.

The client connection attempt can be denied by the server for any reason. The normal procedure will be to have a separate authentication interface (see the companion standards) that is used to authenticate operator and/or physical host computer on the client side. By comparing the session identifier (`M_SESSION` attribute) for the new interface connection

AVA Documents Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11

attempt and the previous authentication transaction, the identity of the client can be established and used to deny or accept the connection attempt.

The death of the local LNA should cause all interfaces to go to the created state, possibly without call-back to the connection handlers. The call-back will be dependent on the API implementation. The death of a remote client MAU will only be reported to the MAU state change handler (see 5.2.4) as a session close event. If necessary, the application should use the session down event to reset the state related to remote clients. The handling of these cases may vary between APIs.

Output parameters: client connection attempts can be denied by returning a non-zero value.

Call-back: not applicable.

Exceptions: not applicable.

Errors: not applicable.

5.6 MCP transactions.

Transactions look somewhat different for the different types of MCPs and whether it is seen from a client or a server. This subclause will be further sub-divided to define the functionality and semantics of each of the main group of transactions. Table 10 summarizes the different transaction types. Transactions will be described for both client and server type MCPs in each subclause. Differences will be noted where appropriate.

Table 10 – Transaction types

Transaction type	Client sends	Server sends	Server later initiates
Function	Data W	Data R	
Read	Request	Data R	
Write	Data W	Acknowledge	
Non-acknowledged write	Data W		
Subscribe	Request	Data R	Data R to all clients
Broadcast subscribe	Request	Data R	Data R to all (unreliable)
ABC subscribe		Data R	Data R to all (unreliable)
Individual subscribe	Data W	Data R	Data R individually

All transactions are associated with one or two data structures: data W and data R. Data W is sent by the client, data R is sent by the server, usually as an effect of the arrival of data W. This is typical of the basic function type transaction. The read transaction has an empty data W from the client (it is a request for data only). The write type transaction contains data from the client to the server and a per transaction acknowledgement, without data, from the server. To save bandwidth, the non-acknowledged write does not allow the server to send an acknowledgement. This group of transactions terminates when the last message has been processed by the receiver.

For subscribe type transactions, the server may send more data R after the first acknowledgement. The subscribe transactions do not terminate until explicitly cancelled, either by a cancel request or by the death of the client or server. Note that the individual subscribe allows the client to send data to the server. This data can be used to configure different replies (also for server initiated transmissions) for different clients. Other subscribe forms send the same data structure with the same contents to all listening clients. Anonymous broadcast (ABC) subscribe does not use the normal client/server paradigm. It allows any node to send to or listen on any other node, without knowing the identity of either.

Documents
Document Sharing Hub

Full document details: DIN EN 61162-401:2002-11