INTERNATIONAL STANDARD

ISO/IEC 24756

First edition
2009-04-01

Information technology — Framework for specifying a common access profile (CAP) of needs and capabilities of users, systems, and their environments

Technologies de l'information — Cadre de définition d'un profil d'accès commun (CAP) des besoins et capacités des utilisateurs, des systèmes et de leurs environnements

Reference number
ISO/IEC 24756:2009(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national 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 and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 24756 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 35, *User interfaces*.

Introduction

Users of various systems in various environments can experience temporary or permanent accessibility difficulties. Potential users of systems need to evaluate whether the systems will be accessible to them in the intended environments in which they will be used. Where accessibility can be insufficient, either due to environmental barriers or poor design, these users can wish to resort to assistive technologies (ATs) to provide the required level of accessibility. Currently, there is no common framework for describing accessibility needs or abilities. This requires each potential user to develop their own evaluation method, and then to investigate and evaluate various systems and ATs using this method. However, due to the lack of an existing method, there might also be a lack of suitable information on the abilities of different systems and ATs, leading to inefficiency, confusion, frustration and a general lack of satisfaction by the user.

A variety of difficulties can be encountered when trying to identify suitable ATs to improve accessibility. Accessibility issues being encountered by potential users can inhibit them from obtaining the required information to identify possible ATs that could help improve their accessibility. Lack of experience with ATs can also affect information technology support staff who attempt to assist these potential users.

The need for accessibility extends to all systems that a proposed user can access. The ability for information gathered regarding accessibility issues and solutions for individual users to be portable across systems and environments is essential. This International Standard introduces a model of accessibility as a basis for understanding access issues with the interactions between users and systems in various environments.

Accessibility is multi-dimensional; existing at multiple levels. The model shows that users and systems must share capabilities of communicating. This International Standard provides a framework to specify a profile of common access capabilities (the CAP) of interactive systems, users, and their environment that are necessary for accessibility to be possible.

The CAP is specified in a top-down manner that provides extensibility to be able to include capabilities at increasingly detailed levels.

Information technology — Framework for specifying a common access profile (CAP) of needs and capabilities of users, systems, and their environments

1 Scope

This International Standard defines a framework for specifying a common access profile (CAP) of needs and capabilities of users, computing systems, and their environments, including access that is supported by assistive technologies. It provides a basis for identifying and dealing with accessibility issues in a standardised manner across multiple platforms. It can be used to evaluate the accessibility of existing systems in particular environments for particular users.

2 Conformance

Specifications for systems and/or system components, including assistive technologies, conform to ISO/IEC 24756 if they conform to Clauses 6 and 7 of this International Standard.

3 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.

ISO 639-3, *Codes for the representation of names of languages — Part 3: Alpha-3 code for comprehensive coverage of languages*

ISO 15924, *Information and documentation — Codes for the representation of names of scripts*

ISO 80000 (all parts), *Quantities and units*

4 Terms and definitions

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

4.1
accessibility
usability of a product, service, environment or facility by people with the widest range of capabilities

NOTE 1 The concept of accessibility addresses the full range of user capabilities and is not limited to users who are formally recognised as having a disability.

NOTE 2 The usability-orientated concept of accessibility aims to achieve levels of effectiveness, efficiency and satisfaction that are as high as possible considering the specified context of use, while paying particular attention to the full range of capabilities within the user population.

[ISO 9241-171:2008, definition 3.2]

4.2
usability
extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use

[ISO 9241-11:1998, definition 3.1]

4.3
assistive technology
AT
hardware or software that is added to or incorporated within a system that increases accessibility for an individual

EXAMPLE Braille displays, screen readers, screen magnification software and eye tracking devices are assistive technologies.

[ISO 9241-171:2008, definition 3.5]

4.4
context of use
users, tasks, equipment (hardware, software, and materials), and the physical and social environments in which a product is used

[ISO 9241-11:1998, definition 3.5]

4.5
handicap
anything that might interfere with the accessibility of interactions between users and systems

5 A model of accessibility

Accessibility involves usable interaction between a user and a system. This interaction takes place within a context of use that includes the system, the user, the user's tasks, and the environment. Figure 1 illustrates the environment in which this interaction takes place. Handicaps are anything that might interfere with the accessibility of interactions between users and systems. A handicap can have one or many sources among the system, user, interaction, and/or environment. This model is “blame-free,” since resolving any handicap to the interaction is more important than attributing blame to the source of the handicap.

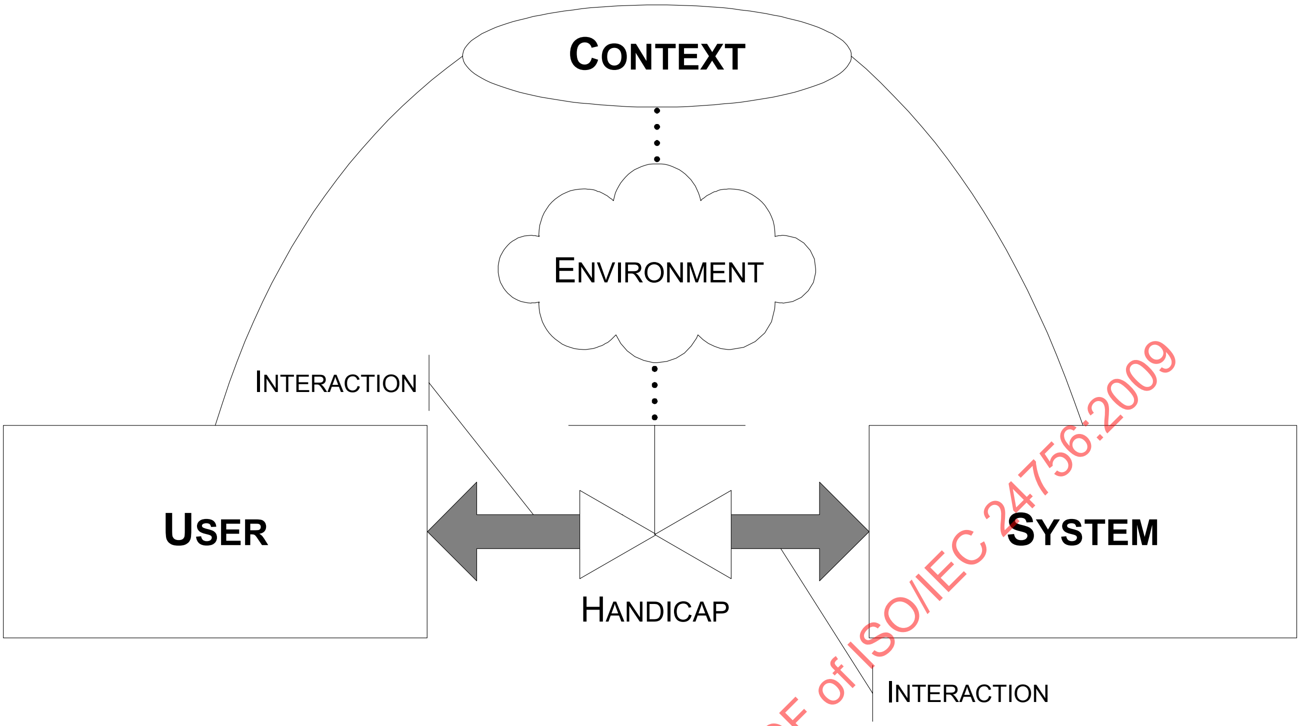

Figure 1 — A model of the User-System interaction

The figure uses a pipe metaphor to illustrate the flow of interactions between the user and the system and a valve metaphor to illustrate various levels of handicaps to the interaction(s). The shaded flow between user and system illustrates the possibility of multiple communications occurring in either direction. A fully open valve represents the absence of a handicap to the interaction. A fully closed valve represents an interaction being fully handicapped. Any other setting of the valve represents an interaction being partially handicapped.

While universal design features can reduce handicaps to interactions, it cannot eliminate all handicaps of the interactions in all situations. An assistive technology (AT) is a means of reducing such handicaps. While a consumer of an AT might not have a disability, there can be some component of the interaction that is “handicapping” them. For example, one could attend a lecture where the speaker uses a language unknown to the listener. Since most people know at least one language, the listener might eventually come to know the language the presentation is given in, but the interaction between speaker and listener is currently handicapped by one not knowing the language used by the other at the present time. The listener’s task of following the details of the presentation would not be possible without the use of a translator to bridge the interaction between the listener and the speaker. In this sense, the translator would be an AT.

Computer related ATs can be realised through: alternative input devices (e.g., trackball, left-handed mouse, sip/puff systems), alternative output devices (e.g., voice, Braille display), accessible software (e.g., screen magnification software), and “universal design” (i.e., barrier-free design). Since the interaction is what is being handicapped, an accessible computing experience is realised by a reduction of this handicap.

ATs can be modelled as a means of opening the valve between systems and users, as shown in Figure 2.

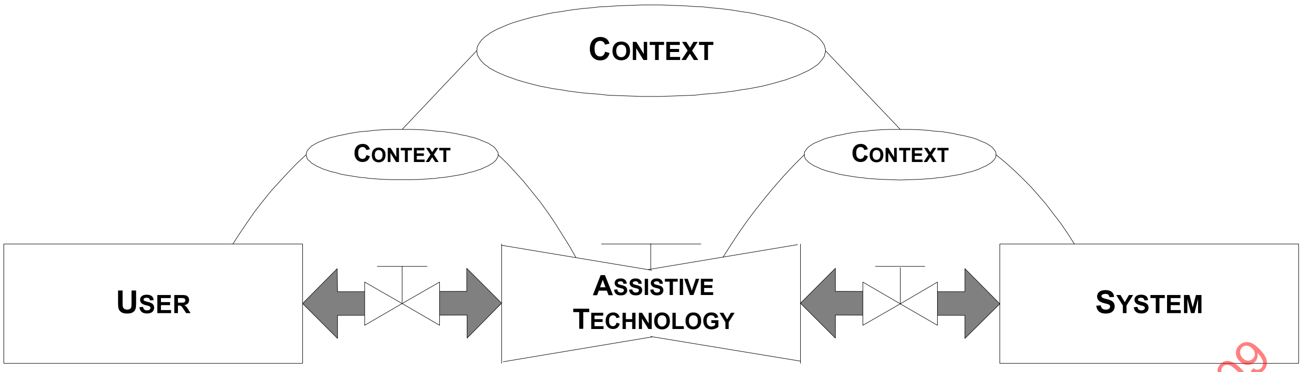

Figure 2 — Assistive Technology in the User-System interaction

Accessibility relies on users and systems using compatible interfaces for interaction. The inclusion of an AT allows translation between two incompatible interfaces as illustrated in Figure 3. To evaluate current and proposed future accessibility, there is a need for a standard method to describe both user-system accessibility and user-AT-system accessibility across all users and systems.

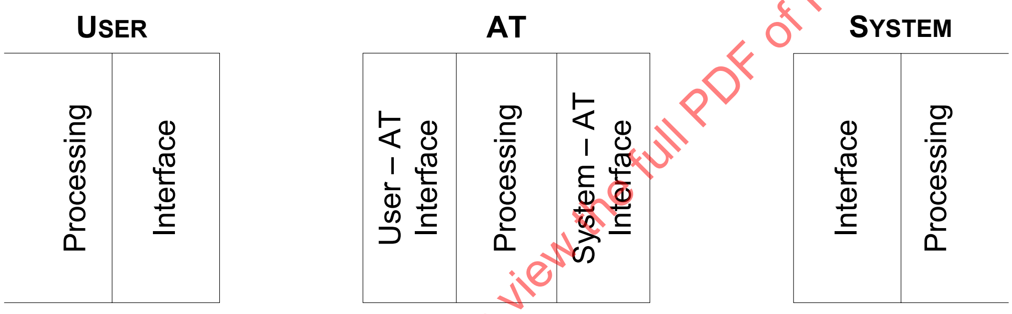

Figure 3 — Interfacing between components

The goal of accessibility is to make systems accessible to users. However, different situations call for different packaging(s) of systems. Where the user's goal is to interact with a particular application package, the user can choose the operating system, computer, peripherals, and other ATs that make the application the most accessible. ATs might be required for accessibility purposes where the user's goal is to interact with an application package that is part of an existing hardware/software system.

The model presented in Figure 3 holds in all situations regardless of the different possible locations of system boundaries. In this model, ATs can be considered anything that is added to the basic system to make it accessible to users. There is a very wide range of objects that can act as ATs, including: special purpose assistive technologies, universal remote consoles, intelligent agents, and even components that are specifically chosen to meet the accessibility needs of a particular user. Multiple ATs can be used in sequence and/or in parallel to support access.

Figure 4 illustrates the paths between the User and their ultimate goal, the application (A1, A2, A3) the user wants to use. Multiple communications can occur in either direction along the connecting lines between components. The applications being used must be accessible to the user. To this end, software-based Assistive Technology (SAT), software which may be part of the operating system or software that is added to the system to increase accessibility for an individual user, might be needed. Examples of SATs include add-on or built-in screen readers.

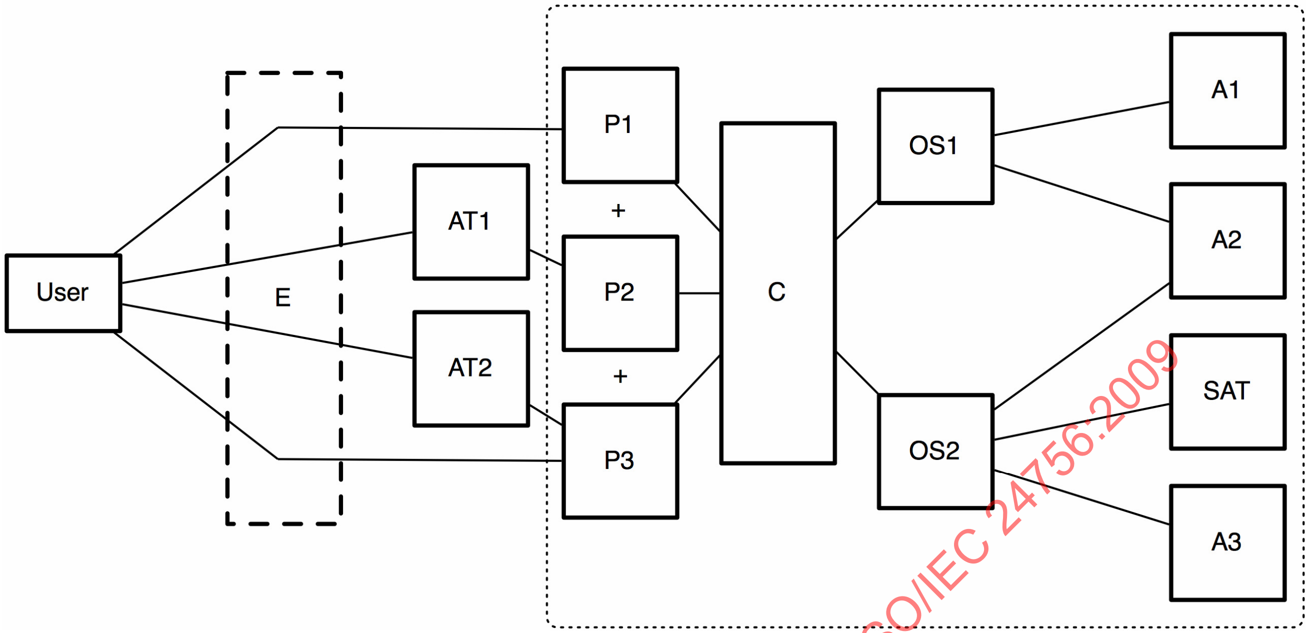

Figure 4 — Components of accessibility

Although not traditionally considered assistive technology, each of the layers (Operating System, Hardware, Peripherals, Assistive Technology, and Environment) between the user and the application has the same effect as an AT in either increasing or decreasing access. The choice of operating system (OS1, OS2) to use with the application can limit or increase the user's access to the application. It can limit access where it does not support certain forms of interaction between the user and the application. It might increase access where it supports transformations of interactions between the user and the application from one form of interaction to another. The computer (C) with which the operating system interacts might limit the user's experience still further. Users are also limited by the capabilities of peripherals (P1, P2, P3) available with the computer.

The user might perceive the combination of application, operating system, computer, and peripherals as a single system, as is indicated by a dotted box in the figure. When considering accessibility, these components can be modelled separately or as a single system.

Assistive Technologies (AT1, AT2) can be used to transform interactions of peripherals to make them more accessible. Environmental conditions (E) can further degrade the accessibility of certain interactions.

To the user, the total experience with all of these components might be perceived as a total system. It is the total system that needs to be specified to evaluate accessibility for the user.

6 A format for identifying access potential

6.1 Introduction to the Common Access Profile

Communications are transmitted (by systems, users, or ATs through channels and environments) to their intended receptors (systems, users, or ATs). This involves flows of information from the system to the user and from the user to the system. The characteristics of these flows are not necessarily the same (e.g., the system might provide spoken output which the user can hear however, if the user has a speech disability, they might choose to use a keyboard to input information to the system). Access exists when the receptor is able to receive and understand the message as transmitted. In this International Standard, systems, users, ATs, environments, and channels will be considered Interacting Components (ICs). Individual communications can be modelled in terms of the receptors, channels, and transmitters used to accomplish the communication. Interaction involves many sets of communications going in either direction between the ICs in the interaction.

An access framework modelling all of the sets of transmitters, channels, and receptors involved in the set of possible interactions between a particular user and a particular system can be used to evaluate the accessibility of a system in a given environment to a particular user.

This access framework involves multiple sets of:

{ Interactions each of which is composed of one or more sets of { receptor, channel, transmitter } }

Rather than deal with each interaction, it is possible to model the set of potential interactions based on an understanding of the compatibility of transmitters, receptors, and channel characteristics of the ICs.

6.2 Common Access Profile

An overall Common Access Profile (CAP_O) is composed of the CAP_{IC} of each different Interacting Component (IC), including those of: users (CAP_{USE}), systems (CAP_{SYS}), assistive technologies (CAP_{AT}), and environments (CAP_{ENV}).

$$(CAP_O) = \Sigma\,(CAP_{IC}) = \begin{aligned} &\text{any } (CAP_{USE}) \cup \\ &\text{any } (CAP_{SYS}) \cup \\ &\text{any } (CAP_{AT}) \cup \\ &\text{any } (CAP_{ENV}) \end{aligned}$$

NOTE 1 The union operation ($\cup$) is used to indicate a composition (collection) of lower-level CAPs pertaining to a CAP. Such compositions are further referenced as “Lower-CAP Linkages” for a specific CAP in this International Standard (see Tables 2 and 3).

The CAP_{IC} of each IC (user, system, AT, environment) is in turn composed of the CAP(s) of each of its Component Features (CAP_{CF}) that provide specifics of various directional communications and processes, these include: the CAP_{IR} of each Input Receptor (IR), the CAP_{OT} of each Output Transmitter (OT), and the CAP_{PF} of each Processing Function (PF) involved in the IC. Describing PFs is optional for users and systems, but is required for ATs.

$$(CAP_{IC}) = \Sigma\,(CAP_{CF}) = \begin{aligned} &\text{any } (CAP_{IR}) \cup \\ &\text{any } (CAP_{OT}) \cup \\ &\text{any } (CAP_{PF}) \cup \end{aligned}$$

ICs can make use of one or more OTs and/or IRs. Where multiple OTs or IRs are required they will be ANDed within the CAP specification. Where substitutions of OTs or IRs are possible, they will be ORed within the CAP specification.

NOTE 2 (IR1 AND IR2) is equivalent to (IR1, IR2).

EXAMPLE (IR1 AND (IR2 OR IR3)) requires that input receptor IR1 always be used and that either input receptor IR2 or input receptor IR3 be used.

Systems are intended to help users to perform tasks. Systems might or might not be directly accessible by users. The CAP of a system provides the starting point for evaluating and improving the accessibility of the system for a user in a given environment. The Environment can reduce the accessibility of a system. ATs might be used to increase the accessibility of a system. Thus, an evaluation of access involves analysing the CAP(s) of a set of systems, users, environments, and ATs.

Figure 5 depicts the structure of the CAP. This four-level structure places CF Type-Specific Information [i.e., modality (CAP_M), capability (CAP_C), and processing (CAP_P)] within their own specific tables. Only those records that are applicable will be coded, leading to simplification and space saving.

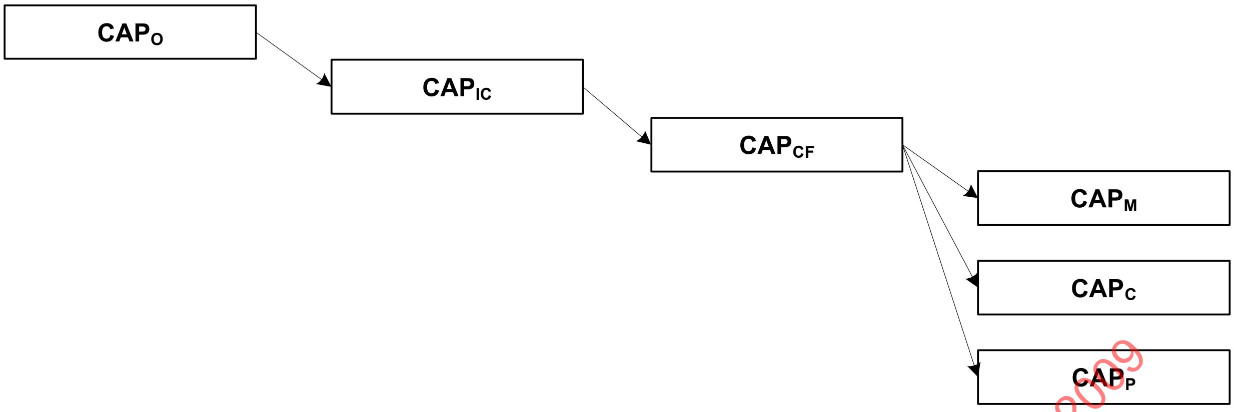

Figure 5 — The CAP structure

6.3 Describing Overall CAPs

The overall CAP_O of a group of ICs shall be specified as outlined in Table 1. Every CAP_O specification has an Identification Information section containing information such as the unique *Name* of the CAP_O, its *Type* (i.e., CAP_O), and a *Qualifier*. It might also contain an unstructured narrative description. Narrative *Descriptions* can be used to record preliminary information and/or provide an easy to read introduction to the structured details of all CAP specifications. All useful CAP_O specifications have linkages to one or more CAP_{IC}(s) and can have linkages to other CAP_O(s).

	Description	*Possible Values*
Identification		
Type	The record type.	CAP_O
Name	An identifier of, or a commonly known name for, the CAP_O.	any (must be unique within CAP)
Qualifier	A unary operator that qualifies this record as being required, optional, or excluded.	one of {SHALL MAY NOT}
Description	A narrative description to record preliminary information and / or optional comments further describing the object.	any
Linkages		
Peer-CAP_O	Peers to this CAP_O.	{<*cap-o-name*, *linkage-type*>, <*cap-o-name*, *linkage-type*>, …}
Lower-CAP_{IC}	The ICs used by this CAP_O.	{<*cap-ic-name*, *linkage-type*>, <*cap-ic-name*, *linkage-type*>, …}

Table 1 — High level CAP_O structure

Linkages are described as <*cap-name, linkage-type*> pairs. The *cap-name* field is the name of the target CAP. The *linkage-type* field describes the applicable binary operator this link implies (i.e. AND, OR, XOR), but if left blank will imply the default linkage type for the given IC type (i.e. AND in the case of CAP_{SYS}/CAP_{AT}/CAP_{ENV}, OR in the case of CAP_{USE}). See 7.3 for more information.

6.4 Describing Interacting Components

Each CAP_{IC} shall be specified as outlined in Table 2. Every CAP_{IC} specification has an Identification Information section with the unique *Name* of the CAP_{IC} and the *Type* (i.e., CAP_{USE}, CAP_{SYS}, CAP_{AT}, or CAP_{ENV} of the IC specification as well as an unstructured narrative *Description* and a *Qualifier*. All CAP_{IC} specifications have linkages to one or more IR/OT/PF Component Feature specifications as well as to the CAP_O(s) to whom it belongs.

	Description	*Possible Values*
Identification		
Type	The record type.	one of {CAP_{USE} CAP_{SYS} CAP_{AT} CAP_{ENV} }
Name	An identifier of, or a commonly known name for, the IC.	any (must be unique within CAP)
Qualifier	A unary operator that qualifies this record as being required, optional, or excluded.	one of {SHALL MAY NOT}
Description	A narrative description to record preliminary information and / or optional comments further describing the object.	any
Linkages		
Higher-CAP_O	The CAP_O to whom this IC belongs.	{<*cap-o-name*, *linkage-type*>, <*cap-o-name*, *linkage-type*>, …}
Peer-CAP_{IC}	Peers to this IC. Linkages to Channel ICs, indicate the number of channel connections to this IC.	{<*cap-ic-name*, *linkage-type*>, <*cap-ic-name*, *linkage-type*>, …}
Lower-CAP_{IR}	The IRs used by this IC.	{<*cap-ir-name*, *linkage-type*>, <*cap-ir-name*, *linkage-type*>, …}
Lower-CAP_{PF}	The PFs used by this IC.	{<*cap-pf-name*, *linkage-type*>, <*cap-pf-name*, *linkage-type*>, …}
Lower-CAP_{OT}	The OTs used by this IC.	{<*cap-ot-name*, *linkage-type*>, <*cap-ot-name*, *linkage-type*>, …}

Table 2 — Interacting Component CAP_{IC} structure

6.5 Describing IC Component Features

Communication is only possible where there are corresponding IRs for the OTs being used. Thus a common format is used to describe both IRs and OTs. Environments can be modelled as components with their own IR and OT and with processing that potentially inhibits access. Processing transforms communications between inputs and outputs and thus is represented by a pair of input and output formats along with a rule to describe the transformation. User and system processing is usually outside the bounds of evaluation. Environmental processing only affects the usability of the communication. AT processing effects the communication by transforming its characteristics.

Each CAP_{IR}, CAP_{OT}, or CAP_{PF} shall be specified as outlined in Table 3. IC Component Feature (CF) specification has an Identification Information section containing information such as the unique *Name* of the CF and the *Type* (i.e., CAP_{IR}, CAP_{OT}, or CAP_{PF}) of CAP_{CF} specification. It might also contain an unstructured narrative *Description* and a *Qualifier*.

	Description	*Possible Values*
Identification		
Type	The record type.	one of {CAP_{IR} CAP_{PF} CAP_{OT}}
Name	An identifier of, or a commonly known name for, the CF.	any (must be unique within CAP)
Qualifier	A unary operator that qualifies this record as being required, optional, or excluded.	one of {SHALL MAY NOT}
Description	A narrative description to record preliminary information and / or optional comments further describing the object.	any
Linkages		
Higher-CAP_{IC}	The IC(s) to whom this CF belongs.	{<*cap-ic-name*, *linkage-type*>, <*cap-ic-name*, *linkage-type*>, …}
Peer-CAP_{IR}	The IRs used by this CF.	{<*cap-ir-name*, *linkage-type*>, <*cap-ir-name*, *linkage-type*>, …}
Peer-CAP_{PF}	The PFs used by this CF.	{<*cap-pf-name*, *linkage-type*>, <*cap-pf-name*, *linkage-type*>, …}
Peer-CAP_{OT}	The OTs used by this CF.	{<*cap-ot-name*, *linkage-type*>, <*cap-ot-name*, *linkage-type*>, …}
Lower-CAP_{M}	The modality specifications of this CF.	{<*cap-m-name*, *linkage-type*>, <*cap-m-name*, *linkage-type*>, …}
Lower-CAP_{C}	The capabilities of this CF.	{<*cap-c-name*, *linkage-type*>, <*cap-c-name*, *linkage-type*>, …}
Lower-CAP_{P}	The processing specifications of this CF.	{<*cap-p-name*, *linkage-type*>, <*cap-p-name*, *linkage-type*>, …}
Connectivity		
Channel capacity	The maximum number of channels the CF can accept.	one of {1 any other specific integer N}
Sharing capability	The need for a CF to have a dedicated channel	one of {SHARABLE DEDICATED POSSIBLE}
CF operations	The amount of time that a CF requires the use of a channel.	one of {INTERMITTENT CONTINUOUS}
Priority	The priority of this CF when using a shared channel	one of {LOW MEDIUM HIGH URGENT}

Table 3 — IC Component Feature CAP_{CF} general format

Each CAP$_{CF}$ specification has a linkage to the CAP$_{IC}$ to which it belongs and can also have linkages to the CAP$_{CF}$ of the other CF(s) to which this CF is directly connected within this IC (e.g., to other IRs/OTs and, as appropriate, to zero, one or more PFs). Linkages to other ICs are handled at the CAP$_{IC}$ level. The CAP$_{CF}$ also includes linkages to three types of specific information that describes the modality, capability, and processing attributes of the CF that are the basis for establishing access.

CAP$_{CF}$s also contain information about the channels they use to connect with each other. Each CF has a *Channel capacity* that indicates the number of channels that it can be connected to at one time. This can be any integer number or the value "N" if it is unlimited in the number of channels. Its *Sharing capacity* indicates whether or not it can share a channel with other similar CFs. A *Sharing capacity* can be "SHARABLE", "DEDICATED" (i.e., not sharable), or "POSSIBLE". A CF’s ability to be shared is based on the *CF operations* and/or the *Priority* of the various CFs that would be involved in the sharing. If the CF operates intermittently, then there will be times when it does not use the channel, and thus the channel could be shared with another intermittently operating CF. However, if one of the CFs involved with the channel operates continuously, then that CF must have a sharing capacity of "SHARABLE" to share the channel. If one of the CFs involved with the channel has a *Priority* of URGENT, then no other CF with a similar priority can share the channel.

6.6 Modality-specific information

Modality-specific information to be linked to a CAP$_{CF}$ shall be specified as outlined in Table 4. This includes identifying the basic modality of a CF, the media, if any, used in this modality of the CF, and the language(s), if any, used in these media. Each CF has a single modality. Multiple modalities of an IC are represented by multiple CF(s) being treated as separate parts of an IC.

Where a PF transforms modalities, the input modality is indicated in the modality specific information and the output modality information is indicated as the result of the transformation.

The *Modality Type* "ALL" means that the CF being described can support all relevant modalities (i.e., visual, auditory, tactile, olfactory). This type is most relevant to user and environment CAPs because it explicitly means that a specific user’s or environment’s capabilities support all modalities and does not need further description. Because it crosses multiple modalities, the *Media Type* "ALL" is allowed if and only if the *Modality Type* "ALL" has been specified.

Identification		
	Description	*Possible Values*
Type	The record type.	CAP$_{M}$
Name	An identifier of, or a commonly known name for, the modality.	any (must be unique within CAP)
Qualifier	A unary operator that qualifies this record as being required, optional, or excluded.	one of {SHALL MAY NOT}
Description	See Table 3.	
Linkages		
Higher-CAP$_{IR}$	The IR(s) to whom this CAP$_{M}$ belongs.	{<*cap-ir-name*, *linkage-type*>, <*cap-ir-name*, *linkage-type*>, …}
Higher-CAP$_{OT}$	The OT(s) to whom this CAP$_{M}$ belongs.	{<*cap-ot-name*, *linkage-type*>, <*cap-ot-name*, *linkage-type*>, …}
Higher-CAP$_{PF}$	The PF(s) to whom this CAP$_{M}$ belongs.	{<*cap-pf-name*, *linkage-type*>, <*cap-pf-name*, *linkage-type*>, …}
Peer-CAP$_{M}$	The other modalities used by this IR/OT/PF.	{<*cap-m-name*, *linkage-type*>, <*cap-m-name*, *linkage-type*>, …}
Peer-CAP$_{C}$	The capabilities used by this IR/OT/PF.	{<*cap-c-name*, *linkage-type*>, <*cap-c-name*, *linkage-type*>, …}
Peer-CAP$_{P}$	The processing specifications used by this IR/OT/PF.	{<*cap-p-name*, *linkage-type*>, <*cap-p-name*, *linkage-type*>, …}

Modality-specific information		
Modality Type	The modality type of this CF. Multiple modalities require separate CF(s)	one of {ALL VISUAL AUDITORY TACTILE OLFACTORY}
Media Types	All the media types, if any, used in this modality by this CF.	{ALL} or {NONE} or one or more of {TextWritten, TextSpoken, TextSigned, TextTactile, Picture, VisualModel, Movie, DynamicVisualModel, Gesture Sound, Music, Texture, TactileGraphic, ForceFeedback, Temperature, Odor}
Language	A list of pairs of all three character ISO 639-3 language code(s) and four character ISO 15924 script code(s) in <*Language Code*, *Script Code*> format that apply to this CF. Keyword "NONE" can be substituted for the language code. Keyword "NIL" can be substituted for the script code.	{<abc, DEFG>, <def, NIL>, <NONE, ABCD>, <NONE, NIL>, …}

Table 4 — Modality-specific information within the CAP_{CF} structure

Modalities can include none, one, or more media. The set of media in Table 5 can be used to identify the various major types of media. This table includes descriptions of each type of media.

Media Type	Description
TextWritten	A language-based medium of words presented in a written symbolic script either statically or dynamically, typically by a system on a screen or by a user on a keyboard.
TextSpoken	A language-based medium of words spoken by the user or system.
TextSigned	A language-based medium of words presented visually in a signed language (e.g., signed video).
TextTactile	A language-based medium of words presented in a tactile symbolic script either statically or dynamically, typically by a system on a tactile display or by a user on a specialized keyboard (e.g., Braille input or output).
Picture	A static image presented by the system or loaded into the system by a user.
VisualModel	An object that combines both data and functionality such that table or model data can be used to render a static image (e.g., graph).
Movie	A dynamic image presented by the system or loaded into the system by a user.
DynamicVisualModel	An object that combines both data and functionality that can be modified by the system or the user and that is presented visually (e.g., animation).
Gesture	Movements of the user that can express an idea or meaning. Gestures can be either tactile or visual in sensation.
Sound	Any media that can be heard by the system or the user but does not necessarily have an associated meaning.

Music	Sounds produced by the system or user arranged in time possessing a degree of melody, harmony, or rhythm.
Texture	Variation of the intensity (feel) of a surface produced by a system or a user such as its smoothness, coarseness, and regularity
TactileGraphic	A static image produced by a system or user presented in a manner appropriate for tactile exploration.
ForceFeedback	A means of tactilely expressing an understanding of the three-dimensional structure, shape, and viscosity of a virtual object.
Temperature	A medium that affects the sense of its degree of hotness or coldness.
Odor	A medium that affects the sense of smell.

Table 5 — Media Types used in a CAP_{CF}

A *Media Type* of "NONE" is used only if the media type is not otherwise obvious. For example, a computer mouse uses the tactile modality but, unless designed to support force feedback, does not use any of the above types of media.

The *Language* entry consists of a list of pairs of <*Language Code, Script Code*> that uniquely identify the languages/scripts used. This scheme ensures that, where multiple languages and scripts are indicated, the “correct” script is always associated with the “correct” language; that is the script(s) used to represent a certain language is paired only with that language. This approach means that one cannot make the mistake of thinking a language (e.g., English) normally represented using one script (e.g., Latin) or its derivatives is represented in another unlikely script (e.g., Cyrillic, Sanskrit, etc.). Therefore, this approach allows both a user to state exactly which language/script pairs are preferred and/or can be used, and content to be explicitly described (e.g., the message contains Ukrainian written using a Cyrillic script and English written using a Latin script). With this approach, an example entry for English written text would be <eng, Latn>. However, some languages might require multiple entries for the same medium (e.g., Japanese written text could require four pairs, one each for Hiragana, Katakana, Kanji, and Latin).

A *Language Code* is a three letter code according to ISO 639-3 that identifies a particular language or the keyword NONE. The keyword NONE was chosen as a four-character keyword to avoid confusion with current and/or future three-character language codes. The keyword NONE is allowed for ICs with no language dependency of any kind (e.g., Environments can be language independent).

A *Script Code* is a four letter code according to ISO 15924 that identifies a particular script or the keyword NIL. Such codes are characterised by the use of four-character *Script Codes*. The keyword NIL was chosen as a three-character keyword to avoid confusion with current and/or future four-character *Script Codes*. A *Script Code* of "NIL" is allowed for media with no script dependency of any kind (e.g., audio text). For simplicity of use and consistency with *Language Code, Script Code* does not use ISO 15924 three-digit codes.

6.7 Capability-specific information

Capability-specific information to be linked to a CAP_{CF} shall be specified as outlined in Table 6. This structure is further expanded for a CAP_P, as described in clause 6.8.

Identification		
	Description	*Possible Values*
Type	The record type.	CAP_C
Name	An identifier of, or a commonly known name for, the capability.	any (unique within the CAP)
Qualifier	A unary operator that qualifies this record as being required, optional, or excluded.	one of {SHALL MAY NOT}
Description	See Table 3.	

Linkages		
Higher-CAP_{IR}	The IR(s) to whom this capability belongs.	{<*cap-ir-name*, *linkage-type*>, <*cap-ir-name*, *linkage-type*>, …}
Higher-CAP_{OT}	The OT(s) to whom this capability belongs.	{<*cap-ot-name*, *linkage-type*>, <*cap-ot-name*, *linkage-type*>, …}
Higher-CAP_{PF}	The PF(s) to whom this capability belongs.	{<*cap-pf-name*, *linkage-type*>, <*cap-pf-name*, *linkage-type*>, …}
Peer-CAP_{M}	The modalities used by this IR/OT/PF.	{<*cap-m-name*, *linkage-type*>, <*cap-m-name*, *linkage-type*>, …}
Peer-CAP_{C}	The other capabilities used by this IR/OT/PF.	{<*cap-c-name*, *linkage-type*>, <*cap-c-name*, *linkage-type*>, …}
Peer-CAP_{P}	The processing specifications used by this IR/OT/PF.	{<*cap-p-name*, *linkage-type*>, <*cap-p-name*, *linkage-type*>, …}
Capability-specific information		
Capability name	The name of the capability. The name "other" is used as the first part of the name for capabilities coming from other specifications beyond this International Standard. A short name or other reference to the specification can be added to the name.	any
Capability instance	To allow multiple capabilities with the same name, an instance number can be used. If no instance number is provided, default is 0 or one higher than the previous *Capability instance* value of the same *Capability name*.	0 (zero or higher number)
Capability values	This information is optional. Either: a) A list of three values of upper and lower values and units (in any unit of measure described in ISO 80000, if applicable) of the *Capability name* for this IR/OT. Assigning the same value as both an upper and lower bound specifies a single value. b) Format dictated by another specification.	{<upper, lower, unit>, <upper, lower, unit>, …} or {any}

Table 6 — Capability-specific information within the CAP_{CF} structure

CAP_{IR}s and CAP_{OT}s document the Capabilities of an IC through the limitations of the capabilities of the user or the system placed on the interactions between users and systems. Interactions are limited or constrained by their need to involve matching capabilities. Usually, these constraints map such that they describe the capabilities of both the user and the system and, as such, can be used to document user capabilities the system needs to match or system capabilities that may help or hinder the user.

Capabilities in a CAP_{CF} refer to the capabilities of a CF within a given modality and media type and/or the range of capabilities possible by the design of a CF. Each CAP_{CF} can have multiple sets of capabilities.

There are three specific fields for documenting capability-specific information in CAP_{IR}s and CAP_{OT}s:

Capability name: Each capability has a capability name value. The name "other" is used as the first part of the name for those capabilities coming from other specifications beyond this standard. A short name or other reference to the external specification can be added to the name (e.g. "other-ISO/IEC 24751").

Capability instance: To enable multiple capabilities with the same name, an instance number can be used. If no instance number is provided, the default value is zero (0). Each new instance of a capability with the same *Capability name* will have its *Capability instance* incremented by one (1).

Capability values: This field is optional. *Capability values* can be formatted in one of two manners:

a) The default format is a list of triples of upper and lower values and, where applicable, units of measure of the capability identified in *Capability name* for this CAP_{IR}/CAP_{OT}. Assigning the same value as both an upper and lower bound specifies a single value. Any unit of measure described in ISO 80000 can be used (e.g., metre, gram, second, ampere, bel, astronomical unit, tonne, pixel, etc.). Where no unit of measure applies, unit is set to "NIL".

b) Where *Capability name* includes the name "other" as the first part of the name for the capability, the format of *Capability values* is dictated by another specification also indicated in the Capability name. This case is described in more detail below.

Frequency is one example of a capability. Audio, visual, and tactile media can be described according to the frequencies used for the input/output. Each CAP_{IR} or CAP_{OT} capable of using such media is also only capable of handling certain frequencies and, as such, might even have multiple documented sets of frequency capabilities. Capabilities in a CAP_{IR} or CAP_{OT} can be used to describe the range of frequencies by content within a given modality and media type. Frequencies are usually expressed in hertz (Hz).

EXAMPLE 1 Humans can usually:

— hear sounds between 64 and 23 000 Hz,

— perceive visible light (i.e., colour) between 384 and 769 THz (note: 1 THz = 1012 Hz), and

— feel frequencies of vibration between 10 and 600 Hz.

However, disabilities can reduce the frequencies that an individual can recognize. Documenting frequency capabilities in a CAP_{IR} or CAP_{OT} can also describe the range of frequencies possible by the limitations of the design of an IC.

EXAMPLE 2 A tactile display may be designed to only output within a certain range of frequency.

EXAMPLE 3 A visual display may be designed to be able to output up to sixteen million colours.

Table 7 provides an example of how frequency capabilities can be specified in a CAP_{IR} or CAP_{OT}. Frequency capabilities are input in pairs of upper and lower frequency bounds. This allows the specification of specific frequencies as well as discrete ranges of frequencies.

Field	*Entry*	*Notes*
Capability name	Frequency	*Name of Frequency capability.*
Capability instance	0	*First instance of a frequency capability in this IR/OT.*
Capability values	{ <upper, lower, Hz>, <upper, lower, Hz>, …}	*A list of pairs of upper and lower frequency bounds in hertz. Assigning the same value as both an upper and lower bound specifies a single frequency.*

Table 7 — Frequency as an example IR/OT Capability-Specific Information

It is possible to have multiple sets of {*Capability name, Capability instance,* and *Capability values*} entries in an individual CAP_{IR} or CAP_{OT}. This allows the documentation of several not necessarily related capabilities for any one IC at once.

For example, in addition to describing frequency capabilities, intensity capabilities might also be relevant. Audio, visual, and tactile media can also be described according to the intensity level of input/output. Each CAP_{IR} or CAP_{OT} capable of using such media is also only capable of handling certain intensities and, as such, can have multiple sets of intensity capabilities. Capabilities in a CAP_{IR} or CAP_{OT} can be used to describe the range of intensities used by content within a given modality and media type and/or the range of frequencies and intensities possible by the design of an IC.

NOTE "Intensity" refers to the magnitude of force or energy used per unit of surface, charge, mass, or time. For example, auditory intensity is analogous to the acoustic notion of volume – the greater the intensity, the "louder" the experience of the stimulus – and is typically measured in decibels. For example, visual intensity is analogous to brightness – the greater the intensity, the "brighter" the experience of the light – and is typically measured in candela.

Documenting intensity capabilities in a CAP_{IR} or CAP_{OT} can describe the range of intensities possible by the limitations of the design of an IC. For example, a PC speaker may be designed to only output sounds at 60 to 70 dB.

Table 8 provides an example of how a set of Capabilities can be specified to document both frequency and intensity capabilities in a CAP_{IR} or CAP_{OT}. Frequency and intensity capabilities are input in pairs of upper and lower frequency / intensity bounds. This allows the specification of specific frequencies / intensities as well as discrete ranges of frequencies / intensities.

Field	*Entry*	*Notes*
Capability name	Frequency	*Name of Frequency capability.*
Capability instance	1	*Second instance of frequency capability in this IR/OT.*
Capability values	{ <upper, lower, Hz>, <upper, lower, Hz>, …}	*A list of pairs of upper and lower frequency bounds in hertz. Assigning the same value as both an upper and lower bound specifies a single frequency.*
Capability name	Intensity	*Intensity capabilities.*
Capability instance	0	*First instance of intensity capability in this IR/OT.*
Capability values	{ <upper, lower, dB>, <upper, lower, dB>, …}	*A list of pairs of upper and lower intensity bounds in decibels. Assigning the same value as both an upper and lower bound specifies a single intensity.*

Table 8 — Frequency and intensity as an example IR/OT Capability-Specific Information

Where *Capability name* includes the name "other" as the first part of the name for the capability, the format of Capability values is dictated by another specification also indicated in the Capability name. In this manner, CAPs can include fields and/or data from other specifications.

For example one capability derived from those discussed in ISO/IEC 24751 is a need for specialised cursor properties such as a specific size, colour, or the presence of cursor tails. Since systems have software that directly support changes to the cursors, this capability can be documented as a system capability.

Table 9 provides an example of how, instead of using the default format of the Capability values field described above, an alternative format, based on an external specification (i.e., ISO/IEC 24751), is used to document cursor capabilities in a CAP_{IR} or CAP_{OT}. The Capability values are input solely through XML code. In the interests of space and readability, only a partial ISO/IEC 24751 specification is shown.

Field	*Entry*	*Notes*
Capability name	other-ISO/IEC24751-cursors	*Cursor capabilities based on ISO/IEC 24751.*
Capability instance	0	*First instance of cursor capability in this IR/OT.*

Capability values	<accessForAll xmlns="http://www.imsglobal.org/xsd/accessforallv1p0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.imsglobal.org/xsd/AccessForAll_v1p0.xsd"> <context identifier="ID000000" lang="en"> <display> <screenEnhance> <screenEnhanceGeneric> <cursorSize usage="preferred" value="0.5"/> <cursorColor usage="preferred" value="ffffffff"/> <cursorTrails usage="preferred" value="0.5"/> </screenEnhanceGeneric> </screenEnhance> </display> </context> </accessForAll>	

Table 9 — Cursor requirements as an example IR/OT Capability-Specific Information

Capabilities can also be derived from those described in other standards that do not have a specific means to encode its specification. For example, ISO 9241-20 recommends, “If a task requires users to make responses … within a limited time in order for that response to be valid (a timeout), the time range should be adjustable by the user, including the option to turn off all timing requirements.” Timeouts can create difficulties in access for many kinds of users. As a result, it is useful to enable systems to document any timeouts and users to describe their own timeout requirements. Because ISO 9241-20 does not have its own specification, there is no need to use the keyword “other” when specifying this capability as shown in Table 10.

Field	*Entry*	*Notes*
Capability name	Timeout	*Name of Timeout capability.*
Capability instance	0	*First instance of timeout capability in this IR/OT.*
Capability values	{ <upper, lower, s>, <upper, lower, s>, …}	*A list of pairs of upper and lower timeout bounds in seconds. Assigning the same value as both an upper and lower bound specifies a single timeout.*

Table 10 — Timeout as an example IR/OT Capability-Specific Information

6.8 Processing-specific information

Processing-specific information to be linked to a CAP_{CF} shall be specified as outlined in Table 11. There are three components: capabilities, connectivity, and transformations.

Identification		
	Description	*Possible Values*
Type	The record type.	CAP_{P}
Name	An identifier or a commonly known name.	any (unique within the CAP)
Qualifier	A unary operator that qualifies this record as being required, optional, or excluded.	one of {SHALL MAY NOT}
Description	See Table 3.	

Linkages		
Higher-CAP$_{IR}$	The IR(s) to whom this Processing Specification belongs.	{<*cap-ir-name*, *linkage-type*>, <*cap-ir-name*, *linkage-type*>, …}
Higher-CAP$_{OT}$	The OT(s) to whom this Processing Specification belongs.	{<*cap-ot-name*, *linkage-type*>, <*cap-ot-name*, *linkage-type*>, …}
Higher-CAP$_{PF}$	The PF(s) to whom this Processing Specification belongs.	{<*cap-pf-name*, *linkage-type*>, <*cap-pf-name*, *linkage-type*>, …}
Peer-CAP$_{M}$	The modalities used by this IR/OT/PF.	{<*cap-m-name*, *linkage-type*>, <*cap-m-name*, *linkage-type*>, …}
Peer-CAP$_{C}$	The capabilities used by this IR/OT/PF.	{<*cap-c-name*, *linkage-type*>, <*cap-c-name*, *linkage-type*>, …}
Peer-CAP$_{P}$	The other processing specifications used by this IR/OT/PF.	{<*cap-p-name*, *linkage-type*>, <*cap-p-name*, *linkage-type*>, …}
Processing-specific information		
Capabilities		
Capability name	The name of the capability. The name "other" is used as the first part of the name for capabilities coming from other specifications beyond this International Standard. A short name or other reference to the specification can be added to the name.	any
Capability instance	To allow multiple capabilities with the same name, an instance number can be used. If no instance number is provided, default is 0 or one higher than the previous *Capability instance* value of the same *Capability name*.	0 (zero or higher number)
Capability values	This information is optional. Either: a) A list of three values of upper and lower values and units (if applicable) of the *Capability name* for this PF. Assigning the same value as both an upper and lower bound specifies a single value. b) Format dictated by another specification.	{<upper, lower, unit>, <upper, lower, unit>, …} or {any}
Connectivity		
Pass through	Content passes through the PF unmodified.	one of {YES, NO}
Transformed	Content is output in modified form.	one of {YES, NO}
Transformations		
Media transformation	The PF transforms the Media.	one of {YES, NO}
Lang transformation	The PF transforms the Language.	one of {YES, NO}

Table 11 — Processing-specific information within the CAP$_{CF}$ structure

CAP$_{P}$s are used by CAP$_{PF}$s to document the Capabilities of an IC by identifying the limits of the capabilities of the user or the system placed on the interactions between users and systems. Usually, these limits map such that they describe the capabilities of both the user and the system and, as such, can be used to document user capabilities the system needs to match or system capabilities that may help or hinder the user.

Unlike a CAP$_{IR}$ or CAP$_{OT}$, capabilities in a CAP$_{PF}$ describe needed changes / limitations to the range of capabilities of the transformed CF. This allows transformed content to meet the needs of users and other ICs who cannot perceive and/or use certain modalities.

Each CAP$_{CF}$ can have multiple sets of capabilities. Capabilities in a CAP$_{CF}$ refer to the capabilities of a CF within a given modality and media type and/or the range of capabilities possible by the design of a CF.

There are three specific fields for documenting Capability-Specific Information in CAP$_{P}$s: *Capability name, Capability instance, and Capability values,* which are used similarly to the manner described in Clause 6.7.

The Connectivity of a CAP_P describes the high-level flow of the input from the IR(s) involved to the output of the OT(s) involved. In particular, Connectivity documents whether the input content stream is modified as it flows through the processing described by the CAP_{PF}. There are two basic connections: *Pass through* which is set to "yes" if the input content stream passes through the PF unmodified, and *Transformed* which is set to "yes" if the input content stream is output in some modified form.

Connectivity can describe four basic transformations:

a) no change (A → A) [*Pass through* only];

b) modify the input (A → B) [*Transformed* only];

c) modify and pass-through the input (A → A, B) [both *Pass-through and Transformed*]; and

d) no output (A → NULL) [none].

Multiple new outputs can be produced by connecting a number of PF(s) together.

EXAMPLE 1 An AT which supports text-to-speech modifies written text input into spoken text output.

a) If the AT is not used, then the connectivity is *Pass through* (i.e., written text output).

b) If the written text is changed into spoken text and only the spoken text is output, then the connectivity is *Transformed* only.

c) If the written text is changed into spoken text and both the written and spoken texts are output, then the connectivity is both *Pass through* and *Transformed*.

d) Finally, if nothing is output, then the connectivity is none.

The Transformations of a CAP_P further describes the details of what CF-Specific Information is being changed. It includes changes of Media and/or Language and/or Capabilities. If the PF transforms the media properties of the content stream, the *Media transformation* field is set to "yes". If the PF transforms the Language properties of the content stream, the *Lang transformation* field is set to "yes". The existence of entries describing Capabilities is understood as a transformation of the capabilities of the content stream by default. Media transformations are further parameterized by PF Capability-Specific attributes.

EXAMPLE 2 An AT, which supports text-to-speech and modifies written text input into spoken text output, represents a media transformation (i.e., visual to auditory). In this case, its Capabilities would at least document both the ability to use text and the ability to output speech thus providing the parameters of the media transformation.

6.9 Expanding the CAP

The CAP structure may be expanded to support additional and/or new functionality. Such properties expand Table 3 by adding additional CAP_{CF}-Specific Information fields and/or linkages to new records describing the property(ies). This is designed to allow future expansion of CAP_{IC}(s). Descriptions of any other properties added to a CAP_{IC} specification should be made available to the users of the specification by both internal documentation (e.g., using the Description field of the new record) and external documentation (i.e., information available through a technical support website).

Examples of such expansions may include additional descriptive information for existing fields (e.g., adding dialect / grapholect information to CAP_{CF} Modality-Specific Information) and the need to add or test new or experimental ICs.

7 Operations on CAPs

7.1 CAP operators

Single argument (unary) logical constructs (i.e., "SHALL", "MAY", "NOT") shall be used to qualify single CAP records as well as groups of records. Double argument (binary) formal logic constructs (i.e., "AND", "OR", and "XOR") may be used to provide a means to describe the operations involved in grouping/combining CAP specifications within a CAP_{IC}.

7.2 Unary operations

CAP unary operators describe the properties of a CAP. Unary operators can occur on individual CAPs. In this sense, unary operators qualify CAPs.

Unary operators qualify the CAP record itself and not its relationships to other CAPs. The Qualifier field of the Identification portion of the CAP is used to document the CAP's applicable unary operator.

7.2.1 Required (SHALL)

To express requirements that must be satisfied, either directly or indirectly, by a system, and/or AT, a CAP is prefixed by the "SHALL" qualifier.

The CAP unary operator "SHALL" expresses that the CAP presents an access requirement that has to be met for full communication to take place. This qualifier is mostly applicable to system and AT CAPs because it denotes what the system or AT needs. These needs are either input requirements (e.g., keyboard input is required) or output requirements (e.g., perceiving visual output is required).

The "SHALL" operator also applies to user CAPs but only in limited cases. For example, "SHALL" can be used to explicitly express a user's language preferences even though other language skills exist.

The "SHALL" operator does not apply to environment CAPs. It is not applied to environment CAPs because, although environments can preclude certain modalities, environments do not require certain forms of interaction.

Within the CAP structure, the "SHALL" operator is implied for systems and ATs.

7.2.2 Optional (MAY)

To express optional requirements that can be satisfied, either directly or indirectly, by a system, and/or AT, a CAP is prefixed by the "MAY" qualifier.

The CAP unary operator "MAY" expresses that the CAP is optional. It might or might not occur. This allows a user some choice in the use of specific ICs. This qualifier is mostly applicable to user CAPs because it denotes what the user is capable of. These capabilities are either system/AT input oriented (e.g., can use keyboard) or system/AT output oriented (e.g., can see video screen).

The "MAY" operator can be applied to system or AT CAPs. It is not applied to system or AT CAPs which describe requirements on the user because the user needs to know what types of interactions will occur without any surprises. On the other hand, it can be used to describe parts of the system or AT that are not required of the user but available for use.

EXAMPLE A mouse connected to a system which can be fully accessed via the keyboard alone would be described with the "MAY" operator since the mouse is not required to use the system.

The "MAY" operator can be applied to environment CAPs because, although environments can preclude certain modalities (which can be documented by their omission), and environments do not require certain forms of interaction, users and systems may still need to interact via a specific modality if it is at all possible due to their own capabilities. The capabilities of a user and/or system might be so dependent on a specific modality, even if it is significantly impacted by the environment, that few alternatives exist. "MAY" does not apply to making choices, which is a binary or greater operation.

Within the CAP structure, the "MAY" operator is implied for users.

7.2.3 Exclusion (NOT)

To express requirements that must not be satisfied, either directly or indirectly, by a system, and/or AT, a CAP is prefixed by the "NOT" qualifier.

The CAP unary operator "NOT" expresses the notion that this CAP "shall not occur". This qualifier is mostly applicable to users and environments because it denotes the environment's limitations and the user's capabilities. This operator expresses where a CF cannot be used. Its use in describing a user or environment will force the system to avoid using a specific type of CF which could lead to access problems.

The "NOT" operator does not usually apply to system or AT CAPs because it is typically not needed. There are some exceptions. One exception is the case of when a system component is connected to the system but not available, as would normally be expected, due to some malfunction (e.g., corrupted drivers, not powered, etc.). The CAP_{IC} describing this component as part of the system would still be in the CAP_{O} because it is still part of the larger system. Using "NOT" allows the CAP_{O} to be updated to show the component is not available even though still connected/detected without deleting the original record.

EXAMPLE A computer mouse connected to a system does not work as expected due to a corrupted driver. Instead of deleting the mouse from the CAP, the Qualifier field of the CAP_{IC} describing the mouse is set to "NOT" to show that, while the mouse is present, it is not functioning as expected.

NOTE It is strongly suggested that care needs to be taken with the use of "NOT". On the one hand, the use of the "NOT" operator could be seen as moving the CAP closer to the medical model approach because the CAP is more oriented to describing the capabilities of systems and users not the lack thereof. As a result, it could be argued that simple omission of a capability is sufficient. On the other hand, the inclusion of "NOT" makes sense logically because the structure of traditional logic includes "NOT". Thus, "NOT" is provided for use in limited situations where necessary, but not using it is preferred.

Within the CAP structure, the "NOT" operator can be applied to environments. Since the CAP is focused on access to the interactions between systems and users, environment CAPs are really only needed where they might preclude potential interaction styles. From the standpoint of the model presented in Clause 5, environment CAPs can use "NOT" because the environment only needs to be discussed/described when it interferes with the interactions between systems and users.

7.3 Binary operations

CAP binary operators describe conjunctions of CAP records within a CAP_{IC}. Using such conjunctions allows simplifying the structure of the CAP when describing all the capabilities of an IC. Such conjunctions provide a means to group specifications within a single user/system/AT/environment CAP_{IC} and document connections between these specifications. Documenting the grouping or connecting of CAP_{IC} specifications will fully profile a system, a user, an environment, or an AT.

NOTE For the purposes of this clause, "content" refers both to information and functionality.

7.3.1 Included (AND)

The CAP binary operator "AND" expresses the notion that both operands of the conjunction are included together for access. That is, "AND" joins requirements. This conjunction is implicitly expressed by CAP linkages within CAP_{IC}s and is the default meaning of such linkages for system, AT, and environment CAPs.

EXAMPLE Consider the linkage necessary between the audio and video portions of a film. If both CAP1 (the audio) and CAP2 (the video) exist, and they are both required by the system, then presumably the conjunction "SHALL CAP1 AND SHALL CAP2" is expressed and both specifications within the system's CAP_{IC} are to be used.

NOTE It is important to take care with the use of "AND", since the possibility of simultaneous use of the component features involved implies the potential for conflict.

7.3.2 Substitutable (OR)

The CAP binary operator "OR" expresses the notion that, for access, either operand is required but both can occur. That is, "OR" combines options. This conjunction is implicitly expressed by CAP linkages within CAP_{IC}s and is the default meaning of such linkages for user CAPs. The "OR" operator can be used by CAPs for users, systems and ATs. The OR operator is not appropriate for environment CAPs because environments cannot be substituted.

The "OR" operator implies that the operand CAPs complement each other – they are substitutable. When IC specifications are described using this conjunction, it means that the content of one can be swapped in favour of the other. While the content represented by each CF in the CAP_{IC} cannot be exactly the same, they might be equivalent. The "OR" conjunction implicitly forbids discordant content – the content represented by each operand cannot be so different that they are not swappable. Thus "OR" is frequently used in the CAP_{IC} to designate equivalent alternatives ensuring that a user who cannot access the content represented by one operand is still able to access equivalent content through another. This allows a user some choice to increase accessibility.

EXAMPLE The audio and caption portions of a film are considered equivalent. They are not the same because they use different media. They are equivalent because each expresses fully the content of the other within its respective medium. Users can choose to use either the captions or audio soundtrack, or both simultaneously if desired. If both CAP1 (the audio) and CAP2 (the captions) exist, and at least one of them is required to be presented by the system, then the conjunction "SHALL CAP1 OR SHALL CAP2" is expressed and one or both specifications within the system's CAP_{IC} can be used.

7.3.3 Mutually exclusive (XOR)

The CAP binary operator "XOR" expresses the mutual exclusion of CAP specifications and ensures that conflicts cannot occur. Traditionally, "XOR" expresses the notion that one of the operands is required, but both can never occur together. Mutual exclusion is often necessary to increase access by reducing channel overload. This conjunction is expressed by CAP linkages within CAP_{IC}s. "XOR" is not a default for any such linkages. "XOR" can be used by CAPs for systems, ATs, and users.

As introduced in clause 6.8, processing performed by ATs can be described in terms of their connectivity.

a) With *Pass through* connectivity, content is transformed into a new medium and both the original content stream and the new alternative content stream are provided to the recipient of the content.

b) Without *Pass through* connectivity, content is transformed into a new medium and the original content stream is silenced – only the new alternative content stream is provided to the recipient.

In the case of ATs, only those ATs without *Pass through* connectivity can use the "XOR" operator. Since, the content is transformed into a new medium and the original content stream is silenced, ATs in transformation mode are the only ATs allowed to use "XOR" because this use closely fits the definition of transformation.

In the case of users, "XOR" allows the explicit description of capability limitations.

EXAMPLE 1 A user unable to perform two actions at the same time, such as a chorded keypress (e.g., press both CONTROL and P keys simultaneously), might group CAPs within a CAP_{IC} that describe keypress capabilities with an "XOR" to exclude their simultaneous use.

EXAMPLE 2 A user might need to control how many different audio channels their hearing needs to attend to and thus use an "XOR" to control the presentation of multiple audio channels to control cognitive overloaded.

Although simultaneous use of two CAPs is not always a requirement when considering whether CAPs are mutually exclusive, mutual exclusion is often necessary to ensure that potentially conflicting content is not transmitted across the same channel simultaneously.

EXAMPLE 3 The soundtrack and descriptive audio portions of a film are typically presented on different (auditory) channels, but, since both provide audiotext, users can only listen to one at a time. If both CAP1 (the soundtrack) and CAP2 (the descriptive audio including soundtrack) exist, and both are required but only one of them is allowed to be presented by the system, then the relationship "SHALL CAP1 XOR SHALL CAP2" is expressed and only one specification within the system's CAP_{IC} is used.

8 Applying the CAP

8.1 Introduction to applying the CAP

CAPs can be used to evaluate handicaps to interactions, select ATs, and manage the use of ATs by systems. To apply a CAP, one must first acquire it.

8.2 Applying the CAP to identifying handicaps

An algorithmic approach can be used when identifying handicaps to interactions using CAPs, as shown in Figure 6.

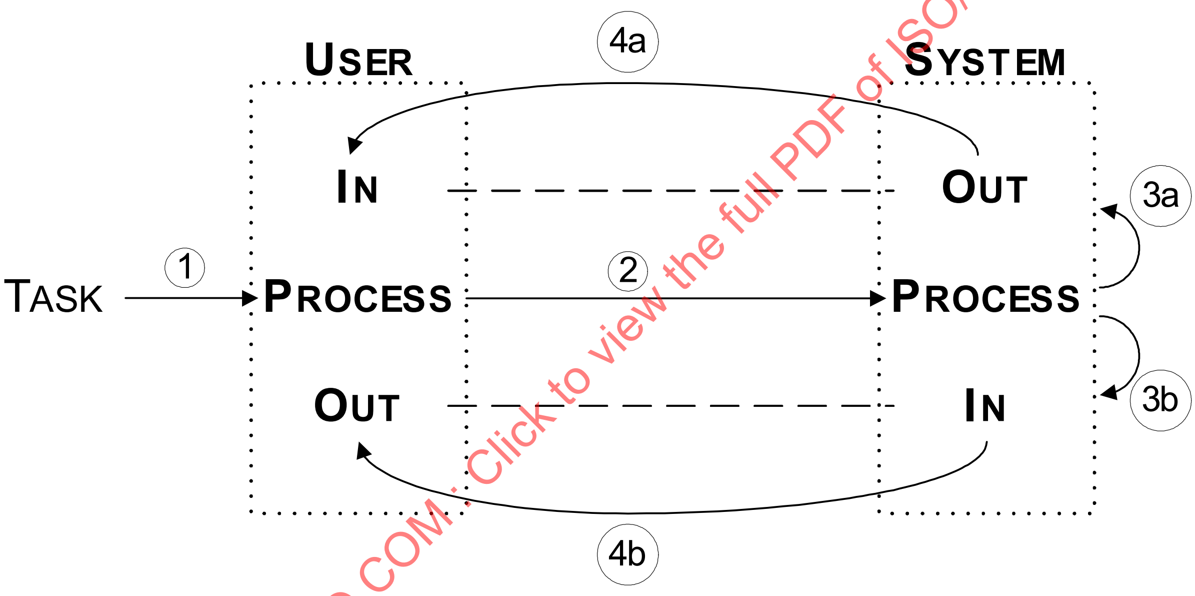

Figure 6 — Identifying handicaps based on the CAP

The starting point is an identified user and system. A user's tasks (1) lead to the selection of specific operating systems and/or application programs to be used. Where particular selections are not made, this defaults to the selection of all known systems, operating systems, and/or applications. These processing requirements are compared (2) to system processing capabilities to select the channels needed for interacting with the user. This involves identifying channels (3a) to send information to the user and identifying channels (3b) to receive user communication. Once these channel selections have been made on the system side, there remains a need to check whether or not the user has the ability to use these channels (4a, 4b). In situations involving multiple users, considerations must be made for each user.

If users do not have the ability to use some subset of these channels, handicaps to the interaction exist. Further handicaps can be identified by the identification of any environmental noise (i.e., auditory, visual, tactile, and/or olfactory noise) and addition of relevant environmental noise CAPs. These CAPs can indicate handicaps of the interaction caused by the environment to channels.

All of the resulting interaction handicaps are opportunities for ATs. Thus, by starting with user and system CAPs, one can identify common needs that are handicapped.

8.3 Applying the CAP to selecting ATs

Given the CAPs of the user, system, and environment; the CAP(s) of potential ATs; and information on interaction handicaps (in terms of channels), an algorithmic approach can be used to identify possibilities to minimise handicaps to the interaction(s).

The set of identified handicaps to the interaction can be used to find an AT that best minimises as many of these handicaps as possible and that interfaces with the system, the user, and the environment. For an AT to interface with the system, it needs to be compatible to the systems' properties (media types provided by operating systems and/or applications). To interface with the user, the AT makes use of skills that the user has (e.g., literacy). To interface successfully, the environment cannot excessively handicap the accessibility of selected channels (e.g., noisy environments can handicap speech output).

Additional information that might be useful in selecting among ATs includes: the time to get the AT working, its cost, and the identification of any benefits to its use. Time needed to get an AT working includes both how long it takes to set up the AT for the initial installation and how long it takes when starting up the system to also start up the AT. Cost refers to the constraint of the consumer's budget and describes sources of funding. Benefits can be identified by a cost/benefit analysis. Desirable benefits include improvements in access and usability.

Once an AT has been selected, its CAP can be added to the CAP of the system. However, the addition of an AT might introduce new handicaps to the interaction (e.g., the best choice might require a capability the system or user does not have) that can require additional ATs. Thus, the process of identifying handicaps to the interaction between the user and the system is repeated after an AT has been added. If there are any handicaps to interactions remaining, interaction handicap minimisation can be repeated until access has reached an acceptable level.

CAPs can also be used to help select combinations of systems and ATs for use in multi-user settings, such as those provided by educational institutions.

8.4 Applying the CAP to managing ATs

CAPs can be used within a system's context to provide the system with information about how best to interface with its user and any ATs being used within the current environment.

CAPs can be created for different base and alternate configurations of users, environments, systems, and ATs. These configuration CAPs can be entered directly into a system or created externally and then downloaded into these systems and/or accessed by these systems when required. They can be further loaded into ATs or accessed by ATs in situations where ATs are programmable.

8.4.1 Developing a base configuration

An initial base configuration of a situation can be described in terms of the set of CAPs representing all relevant systems, ATs, applications, channels, and environments. This requires the identification of the basics that are needed for access and any optimal connections between multiple ATs to be added.

The base configuration must include the most optimal connections among all the ATs. Optimal connections reduce handicaps to the interactions. To identify these connections, the CAPs of each AT will need to be compared to ensure that no two ATs contradict each other. Such contradictions might increase rather than reduce handicaps.

The CAP of an AT can include information about available user capabilities for different channels. This information can assist in the optimal configuration of AT(s) through either AT programmable settings or AT-system configurations.

A base configuration CAP can be developed by selecting CAPs for appropriate system(s) and AT(s) from a database of CAPs and then entering CAPs for the intended user(s) and environment(s). A base configuration can be produced during the selection of ATs or later once the ATs have been procured and installed.

8.4.2 Developing alternate configurations

Once a base configuration has been established, it is possible to develop alternate configurations that respond to changes based on the needs of one or more specific users, applications, systems, ATs, and/or environments.

Differences between users and even within a single user over time provide different access needs. This is especially important in multi-user settings such as those provided by educational institutions where multiple students with different access needs can make use of a limited number of systems and ATs in an accessible lab. Different users have different capabilities as well as different task needs when using the same system at different times. As a result, alternate configurations can be developed for multiple users to share the same system. This allows users who need different AT configurations (including configurations without ATs) to use the same system. Alternate configurations can be developed either proactively or in response to such short-term changes.

Users of a system might utilise specialised CAPs when switching among various applications according to the task they are performing. Alternate CAP configurations can be developed for different applications which make use of different media types with differing levels of usability. The use of specialised application-specific CAPs can increase access to each individual application.

The physical environment within which the user is using the system might not remain stable. Alternate CAP configurations can be developed either proactively or in response to different environments. This allows the system to have ongoing awareness of its environment. The accessibility of mobile computing could benefit from the application of alternate CAP configurations that respond to changes in the environment.

The CAP for a new alternate configuration can start as a copy of the base configuration CAP. This new alternate configuration CAP can then be modified in one or more of the following manners: including additional (system / AT) CAPs from a database of CAPs, entering new and/or modifying existing (system / AT / user / environment) CAPs, and/or deleting (system, AT, user, and/or environment) CAPs that do not apply to the new alternate configuration. There is also the need to be able to delete alternate configurations that no longer apply.

8.4.3 Reconfiguring current configurations

Base and alternate CAP configurations typically change only when the system, AT, or application changes. Such changes result in the need to reconfigure the base and/or alternate CAP configurations. Changes can result from upgrades or replacements to the existing system (including additions or changes to the set of applications), the addition of new ATs, and/or upgrades or replacements to existing ATs.

Changes in the regular environment, and/or permanent changes to the user(s) can also lead to the need to change existing alternate CAP configurations. Differences between users and even within a single user over time provide different access needs. Progressive or permanent changes to one's capabilities mean that each individual user can have different access needs at different times. Such progressive or permanent changes to the user will involve a need to reconfigure their CAP configuration.

For example, some multi-user settings such as those provided by educational institutions are likely to have the need to reconfigure existing configurations regularly based on changes to the user population at least once per term or semester. Ensuring that a system can be changed when needed and can continue to meet the needs of multiple students will achieve the goal of serving the most students with the fewest resources.

Permanent modifications to the system, AT, or application will change existing base and/or alternate CAP configurations. Temporary modifications to the system, AT, application, user, or environment can be accomplished by creating new alternate configurations.

Annex A
(informative)

Example Common Access Profile

A.1 Introduction

The purpose of this Annex is to show how CAPs can be generated using the structure and specification described in Clause 6 and the binary and unary relationships described in Clause 7. For the purposes of this Annex, the olfactory modality will be ignored since it is not currently widely used.

To start these CAPs, we need a CAP_O to contain all the lower level CAPs. Recall that Linkages are described in terms of <*cap-name*, *linkage-type*> pairs and that, where the *linkage-type* is not specified, the default value for the record type is to be assumed. This is shown in Table 12.

CAP_O Identification	
Name	Root
Type	CAP_O
Qualifier	MAY
Description	This is the root node for the examples in this Annex.
CAP_O Linkages	
Peer-CAP_O	
Lower-CAP_{IC}	{<Johanns_Cap, >, <System1, >}

Table 12 — The CAP_O

Table 12 documents the names of the example CAPs discussed in this Annex:

- *Type* contains the default entry for a CAP_O,
- *Qualifier* is "MAY" to indicate the optional use of this CAP_O,
- *Peer-CAP_O* links to nothing because there are currently no peers to this record, and
- *Lower-CAP_{IC}* lists the names of the CAPs discussed in this Annex. Each of these CAPs will be introduced in the discussion below.

A.2 Users

A.2.1 Introduction to User CAPs

This section describes an example user and the contents of their respective CAP_{USE}.

There is no such thing as an "average" user. Similarly, there is no such thing as an "average" CAP_{USE}. As such, one cannot just start building example CAP_{USE}s without having a sense of a specific or generic user and their needs with respect to computer use. For this reason, it is suggested that creation of CAP_{USE}s be guided by the following parameters:

1) If someone other than the user is creating the CAP_{USE}, fully interview the user to determine the capabilities they bring into an interaction with the system.

2) If someone other than the user is creating the CAP_{USE} and, during this interview process, the user identifies specific disabilities, then, to the fullest extent possible, change the statement of disability into a statement of ability.

3) Always assume full support for all modalities and media types unless otherwise stated. Be prepared to document what does work.

4) CAP_{USE} names need to focus on modalities (auditory, visual, tactile, olfactory), not “body parts”. For example use “Visual” rather than “Sight”.

5) CAP_{USE}s must use the data values described in Clause 6.

6) If the ALL keyword is not appropriate (i.e., for ModalityType), consider all of the legal data values. It is possible that one or more of the data values cannot be used. While it may be that some data value will eventually not be included in a specific CAP_{USE}, it must still be considered. Specify each data value that can be used.

7) Capabilities can be described and documented in terms of the user’s senses and other skills as per Clause 6.

8) Because a preference is not an access issue, CAP_{USE}s cannot currently be used to document preferences. Care needs to be taken to ensure that a “requirement” is not a preference in disguise. Such “requirements” are not capabilities but simply preferences based on what a person is used to and thus may be evidence of existing capabilities.

This section provides a brief example user description which focuses on the user’s skills and background. Following the description of the user is an analysis of what input and output modality capabilities the CAP_{USE} for this user would document, example CAP entries, and a high level summary of the user’s CAP_{USE}.

A.2.2 Description of Johann

Well-versed in computers, Johann has little difficulty in their use. Johann wears glasses to correct mild shortsightedness. He is hard of hearing and has trouble hearing error tones if the volume is not sufficiently loud. In addition, he is unable to hear in the left ear and only able to hear sounds within a specific range of volume and pitch in the right. This also limits his ability to work with multiple auditory sources at the same time. He is fluent in German. As a side effect of being hard of hearing, his speech contains some odd inflection and nasality. Johann is dominantly left-handed and tends to prefer using a mouse in the left hand to the point where he will only tolerate using ergonomically right-handed mice in his left hand for short periods.

Given the above description, we can create a CAP_{USE} specifically for Johann. We start by documenting his CAP_{IC} shown in Table 13.

CAP_{IC} Identification	
Name	Johanns_Cap
Type	CAP_{USE}
Qualifier	MAY
Description	This is the starting point of Johann’s user CAP.
CAP_{IC} Linkages	
Higher-CAP_{O}	{<Root, >}
Peer-CAP_{IC}	
Lower-CAP_{IR}	{<Johann_Visual_IR, >, <Johann_Tactile_IR, >, <Johann_Auditory_IR, >,}
Lower-CAP_{PF}	
Lower-CAP_{OT}	{<Johann_OT, >}

Table 13 — A CAP_{USE} for “Johann”

Table 13 documents the start of Johann's CAP_{USE}:

- *Type* contains the default entry for a CAP_{USE}.
- *Qualifier* is "MAY" to indicate the optional use of this user's CAP_{USE} by the CAP_O.
- *Higher-CAP_O* links to the Root CAP_O specified at the beginning of this Annex.
- *Peer-CAP_{IC}* does not link to any peer CAP_{USE}s, since Johann's CAP_{USE} would not interact with other CAP_{USE}s in a CAP_O. Note that this attribute can be later used to link to the CAPs of systems and environments with which Johann interacts.

Because this user does not use all available input modalities, the Lower-CAP_{IR} links to more than one CAP each of which documents Johann's perceptual capabilities in specific modalities.

To document the various CFs that describe Johann, we need CAP_{CF}s that describe all of the modality, capability, and processing information that is relevant to Johann for each of input (CAP_{IR}) and output (CAP_{OT}). Given the above description of this user, his CAP_{CF}s could document the following:

Speech: The user has full speech capabilities. While the odd inflection and nasality reported by this user may impact the use of system input options such as speech-recognition, the user does not report any difficulties with speech-recognition software and thus it is not an issue for the user's CAP_{USE} to record.

Sight: The user reports mild nearsightedness that is corrected with glasses. This user has full visual capabilities.

Hearing: This user reports monaural hearing capability. The user also reports hearing capability within a specific range of volume and pitch. Monaural hearing means that the user cannot be tasked with processing information relying solely on stereo hearing and that any audio input favour the ear that can hear. The user also reports limited ability to work with multiple sources of sound. All of these challenges need to be reflected in this user's CAP_{USE}.

Tactile: This user has full tactile capabilities since there is no report of anything affecting his ability to feel heat and vibration. In addition, this user has full mobility tactile capabilities since there is no report of anything affecting his mobility capabilities. This user also has full dexterity tactile capabilities since there is no report of anything affecting his dexterity capabilities.

The user reports use of ergonomically right-hand-specific devices is difficult and/or uncomfortable due to being dominantly left-handed. Although ergonomically left-handed mice are available, current system support for left-handed users is focused only in setting the primary mouse button of a multi-button mouse to the right-most button.

However, the need for left-handed orientation is a preference. The user is not unable to use their right hand or even to use right-handed input devices; the user is only inconvenienced by them. A preference is not an access issue but an accessibility issue in terms of usability. The CAP currently focuses only on whether access is possible.

A.2.3 A note on hearing

In terms of this user's hearing, no information is provided to determine what range of sounds may need to be amplified (i.e., high or low frequencies). This is expected since audiometric or other medical diagnostic information is not likely to be explicitly known to all users. Two options in this case are to:

- ignore the processing needed to adjust audio output into the user's range of hearing and wait until the user explicitly specifies such an adjustment, or
- specifically encode the CAP describing their hearing using specific assumptions.

Since actions based on the wrong assumption(s) will provide no benefit to the user, the example shown below takes the first option and ignores this issue, only documenting the user's monaural hearing capability. However, two ways to resolve the second option are:

First, the user could be asked if they know what types of sounds they hear well. This assumes the user knows if they have difficulty hearing high frequency sounds, the most common type of sensorineural hearing loss (Nadol, 1993), and/or they have difficulty hearing low frequency sounds. While this approach provides rather poor precision, it may be just enough (or minimally sufficient) to provide better access for the user since it can communicate to the system the need to adjust groups of sounds into perceivable ranges.

Second, if the system was to learn the user's range of hearing through system volume adjustments over time, then information on the minimum and maximum volume (i.e., comfort levels) can be collected and used to compute the user's capabilities and update (with the user's permission) the user's own CAP_{USE}. This is a more adaptive approach which may lead to greater precision for amplification. However, this approach suffers two weaknesses: a) without being combined with the first approach, it may have no impact on the user's perception of all auditory information because it does not adjust the range of frequencies the system uses into a limited group which the user can perceive it only adjusts amplification, and b) it is dependant on the audio hardware being used since user adjustments to hear using one type of audio hardware may not work with another type of audio hardware.

It is not recommended that the system provide some pure tone hearing test to determine the hearing capabilities of the user. The precision of such a test would be strongly influenced by a combination of hardware and environment. Further, this approach may be culturally inappropriate, may take too long for the user's comfort, or may cause discomfort (e.g., tinnitus).

Where AT relevant to the capability is otherwise absent, issues similar to the above discussion may need to be considered in the development of tools designed to support the CAP.

A.2.4 Johann's CAP

This user's CAP_{CF}s document the capability of fully using most available input modalities and specifically document the user's monaural hearing capability. The correct approach to document this is to separately and individually document every one of the user's input capabilities including the user's unique auditory modality input capabilities.

First, we document that this user is capable of fully using visual modality input. Table 14 shows the IR record.

Identification	
Type	CAP_{IR}
Name	Johann_Visual_IR
Qualifier	MAY
Description	Johann's visual modality input capabilities.
Linkages	
Higher-CAP_{IC}	{<Johanns_Cap, >}
Peer-CAP_{IR}	{<Johann_Tactile_IR, >, Johann_Auditory_IR, >,}
Peer-CAP_{PF}	
Peer-CAP_{OT}	{<Johann_OT, >}
Lower-CAP_{M}	{<Johann-Visual_Input_Modality, >}
Lower-CAP_{C}	
Lower-CAP_{P}	
Channels	
Channel capacity	N
Sharing capability	SHARABLE
CF operations	CONTINUOUS
Priority	HIGH

Table 14 — Johann's visual modality input

Table 14 documents Johann's capabilities to perceive visual data:

- *Type* indicates this is an IR record,
- *Qualifier* is "MAY" to indicate the optional use in this user's CAP,
- *Higher-CAP$_{IC}$* links to this user's CAP$_{IC}$,
- *Peer-CAP$_{IR}$* lists the other IR records contained in Johann's CAP because each of these CAPs interacts with the others.
- *Peer-CAP$_{OT}$* lists the only OT record contained in Johann's CAP because this CAP interacts with each of the CAP$_{IR}$s.
- *Lower-CAP$_{M}$* links to a CAP$_{M}$ describing Johann's visual input modality, media, and languages.
- *Channel capacity* is 'N' to show that this user can have multiple visual inputs at a time.
- *Sharing capacity* is SHARABLE because the user is able to work with multiple visual inputs at a time.
- *CF operations* is CONTINUOUS.
- *Priority* is set to HIGH.

The corresponding CAP$_{M}$ appears in Table 15.

Identification	
Type	CAP$_{M}$
Name	Johann-Visual_Input_Modality
Qualifier	MAY
Description	Johann can use any visual input media.
Linkages	
Higher-CAP$_{IR}$	{<Johann_Visual_IR, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	
Peer-CAP$_{C}$	
Peer-CAP$_{P}$	
Modality-Specific Information	
ModalityType	VISUAL
MediaTypes	{TextWritten, Picture, VisualModel, Movie, DynamicVisualModel}
Language	{<deu, Latn>}

Table 15 — Johann's visual modality input media

Table 15 documents the types of media and languages Johann can use in the visual modality:

- *MediaTypes* lists the types of media that are specific to the visual modality.
- *Language* shows this user's language capabilities (in this case, German together with the Latin script).

NOTE *Language*, together with *MediaTypes*'s documentation of support for TextWritten, informs us that Johann can read German text.

Documenting this user's capability to fully use tactile modality inputs includes a similar pair of CAP$_{IR}$ and CAP$_{M}$ records for the tactile modality.

Documenting this user's auditory input capabilities is similar; however it also requires the addition of a CAP$_{C}$ to document this user's unique auditory modality input capabilities. Table 16 shows the IR record.

Identification	
Type	CAP$_{IR}$
Name	Johann_Auditory_IR
Qualifier	MAY
Description	Johann's auditory modality input capabilities.
Linkages	
Higher-CAP$_{IC}$	{<Johanns_Cap, >}
Peer-CAP$_{IR}$	{<Johann_Tactile_IR, >, <Johann_Visual_IR, >,}
Peer-CAP$_{PF}$	
Peer-CAP$_{OT}$	{<Johann_OT, >}
Lower-CAP$_{M}$	{<Johann-Auditory_Input_Modality, >}
Lower-CAP$_{C}$	{<Johann-Auditory_Input_Capability, >}
Lower-CAP$_{P}$	
Channels	
Channel capacity	1
Sharing capability	DEDICATED
CF operations	INTERMITTENT
Priority	HIGH

Table 16 — Johann's auditory modality input

Table 16 documents this user's ability to perceive auditory data:

- *Lower-CAP*$_{C}$ links to a CAP$_{C}$ to list the capabilities of the user's auditory modality input.
- *Channel capacity* is 1 because this user has only one working ear which is tasked for every auditory input event.
- *Sharing capacity* is DEDICATED because the user is unable to work with multiple auditory inputs at a time.
- *Priority* could be set to any value. Technically, it does not apply because the described channel is not SHARABLE.

Table 17 shows the corresponding CAP$_{M}$.

Identification	
Type	CAP$_{M}$
Name	Johann-Auditory_Input_Modality
Qualifier	MAY
Description	Johann can use certain auditory input.
Linkages	
Higher-CAP$_{IR}$	{<Johann_Auditory_IR, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	
Peer-CAP$_{C}$	{<Johann-Auditory_Input_Capability, >}
Peer-CAP$_{P}$	
Modality-Specific Information	
ModalityType	AUDITORY
MediaTypes	{TextSpoken, Sound, Music}
Language	{<deu, NIL>}

Table 17 — Johann's auditory modality input media

Table 17 documents this user's ability to use auditory data:

- *MediaTypes* lists the types of media that are specific to the auditory modality.
- *Language* shows this user's language capabilities (in this case, German).

NOTE Since there is no known script to document German in the auditory modality, *Language* contains the *script-code* 'NIL'.

Table 18 indicates this user's unique auditory modality input capabilities in terms of a CAP_C.

Identification	
Type	CAP_C
Name	Johann-Auditory_Input_Capability
Qualifier	MAY
Description	This capability record documents monaural auditory modality input capabilities.
Linkages	
Higher-CAP_{IR}	{<Johann_Auditory_IR, >}
Higher-CAP_{OT}	
Higher-CAP_{PF}	
Peer-CAP_M	{<Johann-Auditory_Input_Modality, >}
Peer-CAP_C	
Peer-CAP_P	
Capability-Specific Information	
Capability name	Monaural
Capability instance	0
Capability values	Right

Table 18 — Johann's auditory modality input capabilities

The CAP_C in Table 18 documents Johann's ability to use stereo sound:

- *Capability name* "Monaural" enables users to state that of the two or more channels of stereo audio output a system may use, only one can be heard.
- Note no provision is made for specific amplification or minimum volume.

This user has monaural hearing capability and does not have stereo hearing capability. Information encoded in stereo sound cannot be decoded by this user. Table 18 notes this user capability by ensuring all audio output favours the user's right ear.

As shown in Table 19, this user's CAP_{CF}s also document the capability of fully using all available output modalities.

Identification	
Type	CAP_{OT}
Name	Johann_OT
Qualifier	MAY
Description	Johann's output capabilities.
Linkages	
Higher-CAP_{IC}	{<Johann_Cap, >}
Peer-CAP_{IR}	{<Johann_Tactile_IR, >, <Johann_Visual_IR, >, <Johann_Auditory_IR, >,}
Peer-CAP_{PF}	
Peer-CAP_{OT}	
Lower-CAP_{M}	{<Johann-AllModality, >}
Lower-CAP_{C}	
Lower-CAP_{P}	
Channels	
Channel capacity	N
Sharing capability	SHARABLE
CF operations	CONTINUOUS
Priority	URGENT

Table 19 — A CAP for Johann's output capabilities

Table 19 documents Johann's capabilities to produce data:

- *Type* contains the default entry for an OT rather than for an IR,
- *Qualifier* is "MAY" to indicate the optional use in this user's CAP,
- *Higher-*CAP_{IC} links to this user's CAP_{IC},
- *Lower-*CAP_{M} links to the CAP_{M} used by this CAP_{IR}.
- *Peer-*CAP_{IR} links to the peer CAP_{IR} also contained in this user's CAP, and
- *Peer-*CAP_{OT} has no link to any peer CAP_{OT} since this record is the only CAP_{OT}.
- *Priority* is URGENT because the channel settings here apply to all of the output channels. This means that the user can use any output channel at any time since every output channel has the same channel settings.

Since Johann is able to use all available output modalities, rather than documenting each separate modality, we can summarize this information as shown in Table 20.

Identification	
Type	CAP$_{M}$
Name	Johann-AllModality
Qualifier	MAY
Description	Johann's modalities
Linkages	
Higher-CAP$_{IR}$	
Higher-CAP$_{OT}$	{<Johann_OT, >}
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	
Peer-CAP$_{C}$	
Peer-CAP$_{P}$	
Modality-Specific Information	
ModalityType	ALL
MediaTypes	{ALL}
Language	{<deu, Latn>}

Table 20 — Johann's only output modality record

Table 20 documents Johann's supported output modalities:

- *Type* contains the default entry for a CAP$_{M}$,
- *Higher-CAP$_{IR}$* is not set because no CAP$_{IR}$ uses this CAP$_{M}$,
- *Higher-CAP$_{OT}$* links to the CAP$_{OT}$ that uses this CAP$_{M}$,
- *ModalityType* is set to ALL indicating that all modalities are supported by this user,
- *MediaTypes* is set to ALL indicating that all media are supported by this user, and
- *Language* is set to show this user's language capabilities (German together with the Latin script).

If Johann was not able to use all output modalities, then each capability would need to be listed separately, as was the case of his input capabilities. For example, if Johann found it difficult to use software that incorporated timeouts, this would be documented as an output capability necessitating a breakdown of all his other output capabilities. Table 21 shows how a CAP$_{C}$ would document a timeout capability due to dexterity issues.

Identification	
Type	CAP$_{C}$
Name	Johann-Tactile_Output_Capability
Qualifier	MAY
Description	This capability record documents that the user's tactile modality output capabilities restrict use of timeouts.
Linkages	
Higher-CAP$_{IR}$	
Higher-CAP$_{OT}$	{<Johann_Tactile_OT, >}
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	{<Johann-Tactile_Output_Modality, >}
Peer-CAP$_{C}$	
Peer-CAP$_{P}$	
Capability-Specific Information	
Capability name	Timeout
Capability instance	0
Capability values	{<60, 60, s>}

Table 21 — Example timeout CAP$_{C}$

The CAP_C in Table 21 documents how Johann's ability to use software timeouts would be included in his CAP_{USE} if needed:

- *Capability name* "Timeout" enables users to state both the minimum and maximum time they need to respond to a timeout.
- *Capability values* shows that, for this user, a full minute is needed to respond to any timeout. The use of two zeros (i.e., <0, 0, s>) would turn off the use of any timeouts.

The above fully documents Johann's CAP. At a high level, Table 22 describes the structure of Johann's CAP.

Name	Type	Unary Operator	Tables
1.0 Johanns_Cap	CAP_{USE}	MAY	Table 13
1.1 Johann_Tactile_IR	CAP_{IR}	MAY	Like Table 14
1.1.1 Johann-Tactile_Input_Modality	CAP_M	MAY	Like Table 15
1.2 Johann_Visual_IR	CAP_{IR}	MAY	Table 14
1.2.1 Johann-Visual_Input_Modality	CAP_M	MAY	Table 15
1.3 Johann_Auditory_IR	CAP_{IR}	MAY	Table 16
1.3.1 Johann-Auditory_Input_Modality	CAP_M	MAY	Table 17
1.3.2 Johann-Auditory_Input_Capability	CAP_C	MAY	Table 18
1.4 Johann_OT	CAP_{OT}	MAY	Table 19
1.4.1 Johann-AllModality	CAP_M	MAY	Table 20

Table 22 — Structure of CAP_{USE} for "Johann"

Figure 7 shows a high level view of the structure of Johann's CAP in tree form.

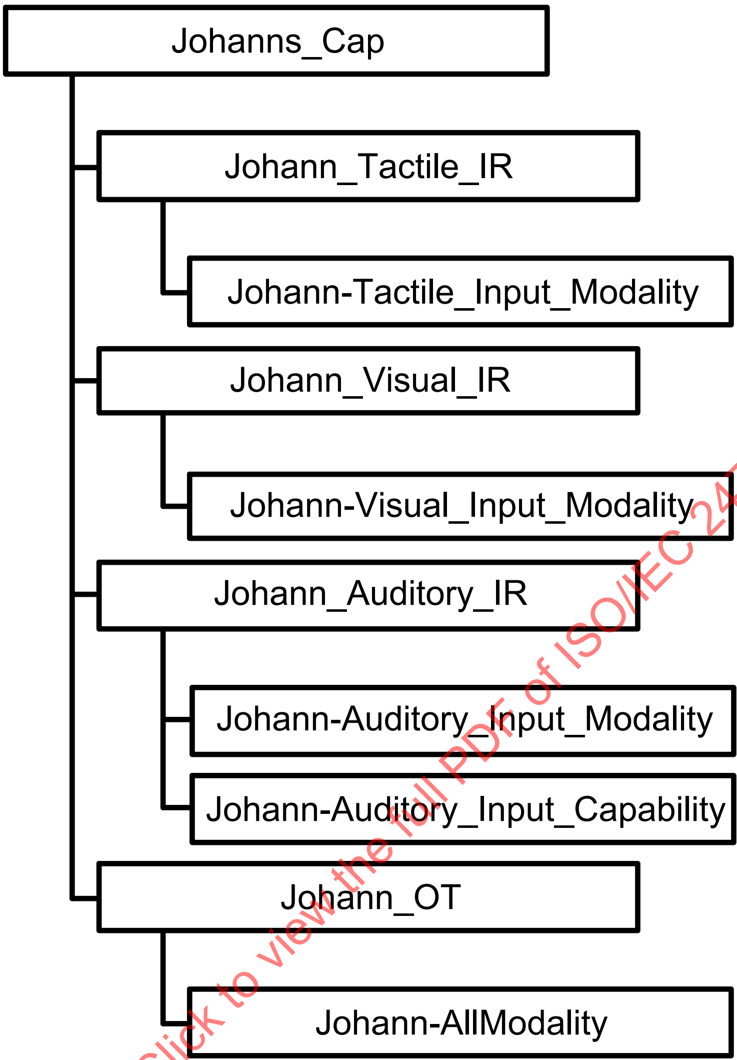

Figure 7 — Tree representation of CAP_{USE} for "Johann"

A.3 Systems

A.3.1 Introduction to System CAPs

This section describes an example system and the contents of its CAP_{SYS}. The description given below is not maximally complete and detailed. CAPs, when used to describe systems, users, and environments can be much more highly detailed. Systems can be described in a CAP_{SYS} in terms of every software capability (e.g., use of graphics, presence of timeouts, capabilities of software-based assistive technologies, etc.). The description given below is intended only for illustration.

It is suggested that creation of CAP_{SYS}s be guided by the following parameters:

1) Approach CAP_{SYS}s as an additive model. Always assume no support for any modalities and media types unless otherwise stated.

2) System capabilities are documented in terms of the supporting hardware and software available.

3) To the greatest extent possible, base CAP_{SYS}s on manufacturer's specifications. Even generic CAP_{SYS}s need some basis in reality.

4) CAP_{SYS} names can focus on modalities as well as systems and their components.

5) CAP_{SYS}s must use the data values described in Clause 6.

6) Because a preference is not an access issue, CAP_{SYS}s cannot be used to document preferences.

Systems are documented within a CAP by documenting all system components including software and peripherals in terms of the modalities they support. For example, a keyboard can be documented in a CAP_{SYS} as a system component that supports tactile input. This approach is similar to how the user example above is documented and has the advantage of allowing devices that have both input and output properties (e.g., a force feedback joystick) to be documented within the same CAP_{SYS} record.

A.3.2 The CAP approach to Systems

This International Standard considers a system to be a combination of hardware and software components that receive input from, and communicate output to, a human user (see ISO 13407). The UARM model presented in Clause 5 presents a system as a composite of processing capabilities, interaction capabilities, and data presented to a user through an interface. In terms of the CAP, this interface can be seen as the capability of a system to receive or transmit information in a specific modality.

For example, a microphone connected to a system is most useful if the system has software that uses it. In this example, "software" encompasses not just hardware drivers but, perhaps more importantly, applications that can use the input auditory modality data received by the microphone. Thus, the microphone is essentially useless if there is no software that uses it.

The example system discussed below presents a CAP which may describe a typical computing system. It highlights the philosophy of ISO 9241-171 to ensure full use via the keyboard only. However, there is no restriction on the use of the mouse.

As noted at the beginning of this Annex, linkages are described in terms of <*cap-name*, *linkage-type*> pairs and, where the *linkage-type* is not specified, the default value for the record type is to be assumed. In the case of systems, AND is the default *linkage-type*. The *linkage-types* OR and XOR must be explicitly stated. Recall that AND implies that both operands of the conjunction "are included together" (not necessarily simultaneously) for access.

A.3.3 Description of Example System

> A desktop computer of the type normally found in many households. This computer has a microphone and speakers that could be used for alternative input/output modalities however, only the already built-in AT consisting of the basic Computer Interface Access Enhancements developed by the Trace R&D Center of the University of Wisconsin-Madison[1)] is available – no other third party provider AT is available.

The system has the following:

Input:

- QWERTY keyboard and software that can read this input.
- Right-handed ergonomic two-button mouse with scroll wheel and software that can use this input.
- Microphone and software that can record or transmit via internet this input.

Output:

- 17-inch monitor capable of supporting all standard resolutions and software that supports this display.
- Stereo speakers and software that outputs various types of audio.

1) That is: StickyKeys, SlowKeys, BounceKeys, FilterKeys, MouseKeys, RepeatKeys, ToggleKeys, SoundSentry, ShowSounds, Time Out and SerialKeys (see ISO 9241-171).

Thus, this system's CAP$_{USE}$ must at least document the following capabilities:

Auditory Input: The presence of a microphone suggests this system supports auditory input capabilities. Its software limits the range of these capabilities to general sounds to be recorded and/or transmitted (e.g., over the internet). This means that the system has sound and music media auditory output capabilities but cannot use this information as input data (e.g., speech recognition).

Auditory Output: The presence of stereo speakers suggests this system supports stereo auditory output capabilities. Its software supports audio playback meaning that the system is capable of playback of any previously recorded audio input. Its software also supports system generated audio output such as error alarms.

Visual Input: The absence of a camera or other visual input device suggests this system does not support visual input capabilities.

Visual Output: The presence of a monitor supporting all standard resolutions suggests this system supports visual output capabilities. Its software fully supports this display. This means that this system is capable of all forms of visual output via the monitor.

Tactile Input: The presence of a keyboard and mouse suggests this system supports some tactile input capabilities. Its software fully supports these input devices. This means that the system is capable of using text input via keyboard and generic mouse input for navigation and manipulation of user interfaces.

Tactile Output: The absence of an explicit tactile output device suggests this system does not support tactile output capabilities.

A.3.4 Example system's CAP

This system's CAP$_{CF}$s document the capability of using specific input and output modalities. We start with the CAP$_{IC}$ for this system, shown in Table 23.

CAP$_{IC}$ Identification	
Name	System1
Type	CAP$_{SYS}$
Qualifier	MAY
Description	This is the starting point of the system CAP for the Example System example.
CAP$_{IC}$ Linkages	
Higher-CAP$_{O}$	{<Root, >}
Peer-CAP$_{IC}$	
Lower-CAP$_{IR}$	{<System1-Tactile_Input, >, <System1-Auditory_Input, OR>}
Lower-CAP$_{PF}$	
Lower-CAP$_{OT}$	{<System1-Visual_Output, >, <System1-Auditory_Output, OR>}

Table 23 — A CAP for "Example System"

Table 23 documents the start of this system's CAP:

- *Higher-CAP$_{O}$* links to the Root CAP$_{O}$ specified at the beginning of this Annex.

In Table 23, the use of the *linkage-type* OR indicates that other modalities are available. For example, the system can support certain types of auditory input. Only those modalities that are ANDed (i.e., the default) indicate modalities that are required by the interface of the system to interact with it.

Now we document all of the input capabilities of this system. If we start with the tactile modality, we can document the keyboard and mouse. The CAP$_{IR}$ for this modality is shown in Table 24.

Identification	
Type	CAP$_{IR}$
Name	System1-Tactile_Input
Qualifier	SHALL
Description	Tactile modality input capabilities. A standard PC US English QWERTY keyboard and a two-button mouse with a scroll wheel.
Linkages	
Higher-CAP$_{IC}$	{<System1, >}
Peer-CAP$_{IR}$	{<System1-Auditory_Input, >}
Peer-CAP$_{PF}$	
Peer-CAP$_{OT}$	{<System1-Visual_Output, OR>, <System1-Auditory_Output, OR>}
Lower-CAP$_{M}$	{<System1-Keyboard_Modality, >, <System1-LeftMouseButton, OR>, <System1-RightMouseButton, OR> <System1-MouseWheel, OR>}
Lower-CAP$_{C}$	{<System1-Keyboard_Capability, >, <System1-Mouse_Capability, OR>}
Lower-CAP$_{P}$	
Channels	
Channel capacity	N
Sharing capability	DEDICATED
CF operations	CONTINUOUS
Priority	HIGH

Table 24 — Tactile input

Table 24 documents the system's capability to use tactile input:

- In this record, the use of the *linkage-type* OR with Lower-CAP$_{M}$ and Lower-CAP$_{C}$ indicates other available tactile modality interfaces and their capabilities, those that are ANDed (i.e., the default) indicate what is needed to use the system.

- As shown in the CAP$_{USE}$ examples, Peer-CAP$_{OT}$s are always ORed in CAP$_{IR}$s (and vice versa). This indicates only what peer CAP$_{OT}$s are available.

While in the real world this may not be the case, this Annex will assume for the purposes of this example that system CFs are defined as simple CFs capable of connecting to only a single channel. In the case of Table 24, this means that the system is only able to receive tactile input from one of the keyboard or mouse, but not both.

Table 25 shows the CAP$_{M}$ for the keys of the keyboard documented in Table 24.

Identification	
Type	CAP_M
Name	System1-Keyboard_Modality
Qualifier	SHALL
Description	The input keys of the keyboard.
Linkages	
Higher-CAP_{IR}	{<System1-Tactile_Input, >}
Higher-CAP_{OT}	
Higher-CAP_{PF}	
Peer-CAP_M	
Peer-CAP_C	{<System1-Keyboard_Capability, >}
Peer-CAP_P	
Modality-Specific Information	
ModalityType	TACTILE
MediaTypes	{TextWritten}
Language	{<eng, Latn>}

Table 25 — Keyboard input modality

Table 25 documents this system's ability to receive text input via keyboard:

- *ModalityType* is TACTILE because physical keyboards represent tactile modality input devices,
- *MediaTypes* indicates that written text is the type of media produced by keyboards, and
- *Language* is set to indicate that this keyboard is specific to Latin alphabets and the English language character set.

Keyboards can come in several sizes and layouts, for example, laptop keyboards are often reduced in size, and BlackBerry™ keyboards are very small in size. Keyboard size is an important access issue for some users. For example, a user who has small hands might not be able to use a full sized keyboard. These dimensions and ISO/IEC 9995-3 recognized national layout (i.e., QWERTY, the US English national layout) can be specified in a CAP_C as shown in Table 26.

We document the mouse starting in Table 27. Note that, according to the record in Table 27, the mouse's buttons and scroll wheel are treated in separate CAP_Ms because they are each operable controls with their own modalities and potential capabilities. Table 27 shows the CAP_M for the left mouse button.

Identification	
Type	CAP_C
Name	System1-Keyboard_Capability
Qualifier	SHALL
Description	The keyboard's capabilities. Here the length, width, height, and layout of the keyboard are described.
Linkages	
Higher-CAP_{IR}	{<System1-Tactile_Input, >}
Higher-CAP_{OT}	
Higher-CAP_{PF}	
Peer-CAP_M	{<System1-Keyboard_Modality, >}
Peer-CAP_C	
Peer-CAP_P	
Capability-Specific Information	
Capability name	Length
Capability instance	0
Capability values	{<45.72, 45.72, cm>}
Capability name	Width
Capability instance	0
Capability values	{<20.32, 20.32, cm>}
Capability name	Height
Capability instance	0
Capability values	{<5.715, 5.715, cm>}
Capability name	other-ISO/IEC 9995-3-KeyboardLayout
Capability instance	0
Capability values	en_US

Table 26 — Keyboard input modality capabilities

Identification	
Type	CAP_M
Name	System1-LeftMouseButton
Qualifier	MAY
Description	The modality of the left-side mouse button.
Linkages	
Higher-CAP_{IR}	{<System1-Tactile_Input, >}
Higher-CAP_{OT}	
Higher-CAP_{PF}	
Peer-CAP_M	{<System1-RightMouseButton, OR>, <System1-MouseWheel, OR>}
Peer-CAP_C	{<Mouse_Capability, >}
Peer-CAP_P	
Modality-Specific Information	
ModalityType	TACTILE
MediaTypes	{NONE}
Language	{<NONE, NIL>}

Table 27 — Left mouse button input modality

Table 27 documents that this mouse button can receive tactile input:

- *Qualifier* is MAY to indicate the optional use of the mouse.
- The *linkage-type* OR is specified on Peer-CAP_Ms at this level to indicate other mouse buttons could also be used,
- The *linkage-type* AND is implied on Peer-CAP_Cs at this level to indicate the capabilities associated with this CAP_M,
- *ModalityType* is set to TACTILE because mice represent a tactile input method,
- *MediaTypes* is set to NONE because the mouse does not support any of the explicit CAP media types listed in Table 5, and
- *Language* is set to indicate that mice are language neutral.

Since the right-hand mouse button and the scroll wheel have the same settings, their records will not be shown here.

Mice may be ergonomically designed to fit a certain hand. ISO/IEC 24751 allows us to specify whether an input device is explicitly ergonomically configured for a right- or left-handed person.

Mice come in several sizes, for example, mini (extra small) mice are commonly used with laptops. Mouse size can impact one user across different ages as well as different users with different handsizes. Since, for this user, a specific mouse size would be a preference, such information was not provided in the user example above. However, mouse size can be documented in a system's CAP to enable users who do not have broad mouse size capabilities to choose mice within a specific size range.

Since there is no current International Standard that provides an explicit definition of pointing device sizes, this information must be encoded using SI units. The actual physical size of the mouse is not easily comparable to users, thus to make this specification even more useful when comparing users, the handsize the mouse is designed for is specified instead. Handsize is measured from the crease of the wrist closest to the hand out to the end of the middle finger.

Thus, Table 28 shows what the CAP_C for this mouse might look like. Note that, in this Annex, all examples using ISO/IEC 24751, in the interests of space and readability, only show the portion of the XML code needed.

Identification	
Type	CAP$_{C}$
Name	Mouse_Capability
Qualifier	MAY
Description	Additional properties and capabilities of the mouse. Here the size and handedness of the mouse is described.
Linkages	
Higher-CAP$_{IR}$	{<System1-Tactile_Input, >}
Higher-CAP$_{OT}$	
Higher-CAP$_{PF}$	
Peer-CAP$_{M}$	{<System1-LeftMouseButton, >, <System1-RightMouseButton, OR>, <System1-MouseWheel, OR>}
Peer-CAP$_{C}$	
Peer-CAP$_{P}$	
Capability-Specific Information	
Capability name	Handsize
Capability instance	0
Capability values	{<17, 19, cm>}
Capability name	other-ISO/IEC24751-handedness
Capability instance	0
Capability values	… <alternativePointing> <handedness value="right"/> </alternativePointing> …

Table 28 — An ergonomically right-handed mouse for a medium sized hand

Table 28 documents this mouse's capabilities:

- The *linkage-type* OR may or may not be specified on Peer-CAP$_{M}$s at this level. If they are ANDed to this CAP$_{C}$, it shows that they go together. If they are ORed, it indicates alternatives.
- The mouse Capability handsize indicates this mouse is considered "medium sized", and
- The handedness Capability documents that this mouse is ergonomically right-handed.

Note that this CAP does not force the user to use all of the mouse buttons. It only documents the presence of the mouse and the number of buttons available. The rightmost mouse button has been ORed to show that back-clicking (in this case, via right-click) is available but not the primary mode of interaction with the mouse (i.e., the leftmost button). This is important because of the access issues (e.g., due to repetitive strain injury, sprained wrist, etc.) which can be associated with using a mouse (e.g., Fagarasanu & Kumar, 2003).

The only other input modality supported by this system is the auditory modality via the microphone and its supporting software. The CAP$_{IR}$ is shown in Table 29.