ETSI GR ARF 001 V1.1.1 (2019-04)

Augmented Reality Framework (ARF); AR standards landscape

Disclaimer

Reference

DGR/ARF-001

Keywords

augmented reality, interoperability, platforms

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) Augmented Reality Framework (ARF).

The ISG ARF shares the following understanding for Augmented Reality: Augmented Reality (AR) is the ability to mix in real-time spatially-registered digital content with the real world. The present report provides a snapshot of standardization efforts conducted by various Standards Development Organizations (SDOs) and other fora, as available at the time of publishing. While the goal of the present document is to provide an exhaustive list of relevant standards for AR, this may not be the case with the first version of the present document and updates may be available in future versions.

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document aims to identify the role of existing standards relevant to augmented reality and to contribute to identify any interoperability gaps. The activity summarized in the present document consisted in analysing the standardization work related to augmented reality in various standards setting organizations. While some of these standards under review are directly addressing AR as a whole, others are addressing key technological components that can be useful to increase interoperability of AR applications and services.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ISO/IEC 18039:2019: "Mixed and augmented reality (MAR) reference model".

NOTE: Available at https://www.iso.org/standard/30824.html.

[i.2] ISO/IEC 23000-13:2017: "Multimedia application format (MPEG-A) -- Part 13: Augmented reality application format" a.k.a ARAF.

NOTE: Available at https://www.iso.org/standard/69465.html.

[i.3] ISO/IEC 14496-1:2010: "Coding of audio-visual objects -- Part 1: Systems".

NOTE: Available at https://www.iso.org/standard/55688.html.

[i.4] ISO/IEC 14496-2:2004: "Coding of audio-visual objects -- Part 2: Visual".

NOTE: Available at https://www.iso.org/standard/39259.html.

[i.5] ISO/IEC 14496-3:2009: "Coding of audio-visual objects -- Part 3: Audio", a.k.a. AAC.

NOTE: Available at https://www.iso.org/standard/53943.html.

[i.6] ISO/IEC 14496-10:2014: "Coding of audio-visual objects -- Part 10: Advanced Video Coding", a.k.a AVC.

NOTE: Available at https://www.iso.org/standard/66069.html.

[i.7] ISO/IEC 14496-11:2015: "Scene description and application engine", a.k.a. BIFS.

NOTE: Available at https://www.iso.org/standard/63548.html.

[i.8] ISO/IEC 14496-16 2011: "Coding of audio-visual objects -- Part 16: Animation Framework eXtension (AFX)".

NOTE: Available at https://www.iso.org/standard/57367.html.

[i.9] ISO/IEC 14772-1:1997: "The Virtual Reality Modeling Language", a.k.a VRML.

NOTE: Available at https://www.iso.org/standard/25508.html.

[i.10] ISO/IEC 19775:2013: "Extensible 3D (X3D)".

NOTE: Available at https://www.iso.org/standard/60760.html.

[i.11] ISO/IEC 23005-5:2016: "Media context and control", a.k.a. MPEG-V.

NOTE: Available at https://www.iso.org/standard/65398.html.

[i.12] ISO 19005-3:2012: "Document management -- Electronic document file format for long-term preservation -- Part 3: Use of ISO 32000-1 with support for embedded files (PDF/A-3)", a.k.a PDF.

NOTE: Available at https://www.iso.org/standard/57229.html.

[i.13] ISO/IEC 10918:1997: "Digital compression and coding of continuous-tone still images", a.k.a. JPEG.

NOTE: Available at https://www.iso.org/standard/25037.html.

[i.14] ISO/IEC 15948 2004: "Portable Network Graphics (PNG)".

NOTE: Available at https://www.iso.org/standard/29581.html.

[i.15] ISO/IEC 15444-2:2004: "JPEG 2000 image coding system".

NOTE: Available at https://www.iso.org/standard/33160.html.

[i.16] ISO/IEC 23008-2:2017: "High efficiency coding and media delivery in heterogeneous environments", a.k.a HEVC.

NOTE: Available at https://www.iso.org/standard/69668.html.

[i.17] ISO/IEC 23090:2019: "Coded representation of immersive media".

NOTE: Available at https://www.iso.org/standard/73310.html.

[i.18] ISO/IEC 11172-3:1993: "Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s", a.k.a mp3.

NOTE: Available at https://www.iso.org/standard/22412.html.

[i.19] ISO/IEC 23003:2018: "MPEG audio technologies -- Part 2: Spatial Audio Object Coding (SAOC)".

NOTE: Available at https://www.iso.org/standard/73122.html.

[i.20] ISO/IEC 23090-5 (under development): "Coded representation of immersive media -- Part 5: Video-based Point Cloud Compression".

NOTE: Available at https://www.iso.org/standard/73025.html.

[i.21] ISO/IEC 23090-9 (under development): "Coded representation of immersive media -- Part 5: Graphics-based Point Cloud Compression".

[i.22] ISO/IEC 30113-1:2015: "User interface -- Gesture-based interfaces across devices and methods".

NOTE: Available at https://www.iso.org/standard/53233.html.

[i.23] ISO/IEC 23007-2:2012: "Rich media user interfaces -- Part 2: Advanced user interaction (AUI) interfaces".

NOTE: Available at https://www.iso.org/standard/59241.html.

[i.24] ISO 55000:2014: "Asset management -- Overview, principles and terminology".

NOTE: Available at https://www.iso.org/standard/55088.html.

[i.25] ISO 55001:2014: "Asset management -- Management systems - Requirements".

NOTE: Available at https://www.iso.org/standard/55089.html.

[i.26] ISO 55002:2018: "Asset management -- Management systems -- Guidelines for the application of ISO 55001".

NOTE: Available at https://www.iso.org/standard/70402.html.

[i.27] ISO 15926-1:2004: "Industrial automation systems and integration -- Integration of life-cycle data for process plants including oil and gas production facilities -- Part 1: Overview and fundamental principles".

NOTE: Available at https://www.iso.org/standard/29556.html.

[i.28] ISO 15926-2:2003: "Industrial automation systems and integration -- Integration of life-cycle data for process plants including oil and gas production facilities -- Part 2: Data model".

NOTE: Available at https://www.iso.org/standard/29557.html.

[i.29] ISO 9241:1983: "Series of standards related to Ergonomics of human-system interaction".

NOTE: Available at https://www.iso.org/committee/53372.html.

[i.30] ISO 9126:2001: "Software engineering -- Product quality -- Part 1: Quality model".

NOTE: Available at https://www.iso.org/standard/22749.html.

[i.31] ISO/IEC 25060:2010: "Systems and software engineering -- Systems and software product Quality Requirements and Evaluation (SQuaRE) -- Common Industry Format (CIF) for usability: General framework for usability-related information".

NOTE: Available at https://www.iso.org/standard/35786.html.

[i.32] ISO/IEC 25062:2006: "Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Common Industry Format (CIF) for usability test reports".

NOTE: Available at https://www.iso.org/standard/43046.html.

[i.33] ISO/IEC 25063:2014: "Systems and software engineering -- Systems and software product Quality Requirements and Evaluation (SQuaRE) -- Common Industry Format (CIF) for usability: Context of use description".

NOTE: Available at https://www.iso.org/standard/35789.html.

[i.34] ISO/IEC 25064:2013: "Systems and software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Common Industry Format (CIF) for usability: User needs report".

NOTE: Available at https://www.iso.org/standard/35790.html.

[i.35] ISO 14915:2002: "Software ergonomics for multimedia user interfaces -- Part 1: Design principles and framework".

NOTE: Available at, https://www.iso.org/standard/25578.html.

[i.36] ISO/IEC 40500:2012: "W3C Web Content Accessibility Guidelines (WCAG) 2.0".

NOTE: Available at https://www.iso.org/standard/58625.html.

[i.37] ISO/IEC 15938-13:2015: "Compact Descriptors for Visual Search".

NOTE: Available at https://www.iso.org/standard/65393.html.

[i.38] ISO/IEC 16739:2018: "Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries -- Part 1: Data schema".

NOTE: Available at https://www.iso.org/standard/70303.html.

[i.39] Recommendation ITU-T H.264 (2017): "Advanced video coding for generic audiovisual services".

NOTE: Available at https://www.itu.int/rec/T-REC-H.264-201704-I/en.

[i.40] Recommendation ITU-T H.265 (2018): "High efficiency video coding".

NOTE: Available at https://www.itu.int/rec/T-REC-H.265-201802-I/en.

[i.41] IETF RFC 7159 (2014): "The JavaScript Object Notation (JSON) Data Interchange Format".

NOTE: Available at https://tools.ietf.org/html/rfc7159.

[i.42] IETF RFC 1889 (1996): "RTP: A Transport Protocol for Real-Time Applications".

NOTE: Available at https://tools.ietf.org/html/rfc1889.

[i.43] IETF RFC 3550 (2003): "RTP: A Transport Protocol for Real-Time Applications".

NOTE: Available at https://tools.ietf.org/html/rfc3550.

[i.44] IETF RFC 2326 (1998): "Real Time Streaming Protocol (RTSP)".

NOTE: Available at https://www.ietf.org/rfc/rfc2326.txt.

[i.45] IETF RFC 7826 (2016): "Real-Time Streaming Protocol Version 2.0".

NOTE: Available at https://tools.ietf.org/html/rfc7826.

[i.46] IETF RFC 5870 (2010): "A Uniform Resource Identifier for Geographic Locations ('geo' URI)".

NOTE: Available at https://tools.ietf.org/html/rfc5870.

[i.47] IETF RFC 7946 (2016): "The GeoJSON Format".

NOTE: Available at https://tools.ietf.org/html/rfc7946.

[i.48] IETF RFC 793 (1981): "Transmission Control Protocol".

NOTE: Available at https://tools.ietf.org/html/rfc793.

[i.49] OGC ARML (2015): "Augmented Reality Markup Language 2.0 (ARML 2.0)".

NOTE: Available at http://docs.opengeospatial.org/is/12-132r4/12-132r4.html.

[i.50] OGC GML (2007): "OpenGIS® Geography Markup Language (GML) Encoding standard".

NOTE: Available at https://www.opengeospatial.org/standards/gml.

[i.51] OGC CityGML (2012): "OGC City Geography Markup Language (CityGML) Encoding Standard".

NOTE: Available at https://www.opengeospatial.org/standards/citygml.

[i.52] OGC IndoorGML (2018): "OGC® IndoorGML Encoding Standard".

NOTE: Available at https://www.opengeospatial.org/standards/indoorgml.

[i.53] OGC KML (2015): "OGC KML 2.3".

NOTE: Available at https://www.opengeospatial.org/standards/kml.

[i.54] OGC WMS (2006): "OpenGIS® Web Map Server Implementation Specification".

NOTE: Available at https://www.opengeospatial.org/standards/wms.

[i.55] W3C WebXR (2019): "WebXR Device API".

NOTE: Available at https://www.w3.org/TR/webxr/.

[i.56] W3C Recommendation 16 August 2006: "Extensible Markup Language (XML) 1.1 (Second Edition)".

NOTE: Available at https://www.w3.org/TR/2006/REC-xml11-20060816/.

[i.57] W3C Working Draft 18 October 2018: "HTML 5.3".

NOTE: Available at https://www.w3.org/TR/html53/.

[i.58] W3C Candidate Recommendation 27 September 2018: "WebRTC 1.0: Real-time Communication Between Browsers".

NOTE: Available at https://www.w3.org/TR/webrtc/.

[i.59] W3C Recommendation 16 August 2011: "Scalable Vector Graphics (SVG) 1.1 (Second Edition)".

NOTE: Available at https://www.w3.org/TR/SVG11/.

[i.60] W3C Candidate Recommendation 3 October 2017: "Media Capture and Streams".

NOTE: Available at https://www.w3.org/TR/mediacapture-streams/.

[i.61] W3C Recommendation 8 November 2016: "Geolocation API Specification 2nd Edition".

NOTE: Available at https://www.w3.org/TR/geolocation-API/.

[i.62] Khronos, COLLADA (2008): "COLLADA Digital Asset Schema Release 1.4.1 Specification (2nd Edition)".

NOTE: Available at https://www.khronos.org/files/collada_spec_1_4.pdf.

[i.63] Khronos, glTF (2017): "GL Transmission Format (glTF) (Version 2.0)".

NOTE: Available at https://github.com/KhronosGroup/glTF/tree/master/specification/2.0.

[i.64] Khronos, OpenVX (2017): "The OpenVX Specification".

NOTE: Available at https://www.khronos.org/registry/OpenVX/specs/1.2/html/index.html.

[i.65] Khronos, OpenXR, under development: "OpenXR Overview".

NOTE: Available at https://www.khronos.org/openxr.

[i.66] Khronos, OpenGL (2019): "The OpenGL© Graphics System, OpenGL 4.6 Core Profile".

NOTE: Available at https://www.khronos.org/registry/OpenGL/specs/gl/glspec46.core.pdf.

[i.67] Khronos, WebGL (2018): "WebGL 2.0 Specification".

NOTE: Available at https://www.khronos.org/registry/webgl/specs/latest/2.0/.

[i.68] Khronos, Vulkan (2019): "Vulkan 1.1 API Specifications".

NOTE: Available at https://www.khronos.org/registry/vulkan/#apispecs.

[i.69] DIN SPEC 91333 (2016): "Contactless gesture control for human-system interaction".

NOTE: Available at https://www.beuth.de/en/technical-rule/din-spec-91333/255376761.

[i.70] DIN EN ISO 9241-960 (2018): "Ergonomics of human-system interaction - Part 960: Framework and guidance for gesture interactions (ISO 9241-960:2017)".

NOTE: Available at https://www.beuth.de/en/standard/din-en-iso-9241-960/272433976.

[i.71] ASTM E2132 - 17 (2017): "Standard Practice for Inventory Verification: Electronic and Physical Inventory of Assets".

NOTE: Available at https://www.astm.org/Standards/E2132.htm.

[i.72] ASTM E3035 - 15 2015: "Standard Classification for Facility Asset Component Tracking System (FACTS)".

NOTE: Available at https://www.astm.org/Standards/E3035.htm.

[i.73] ASTM E2499 - 18 2018: "Standard Practice for Classification of Asset Location Information".

NOTE: Available at https://www.astm.org/Standards/E2499.htm.

[i.74] MIMOSA CCOM 4.0.0 2016: "Open Standards for Physical Asset Management".

NOTE: Available at http://www.mimosa.org/specifications/ccom-4-0-0/.

[i.75] MIMOSA OSA-EAI 2014: "Open Standards for Physical Asset Management".

NOTE: Available at http://www.mimosa.org/mimosa-osa-eai/.

[i.76] Recommendation ITU-T T.81: "Information technology - Digital compression and coding of continuous-tone still images - Requirements and guidelines".

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AAC	Advanced Audio Coding
ADAS	Advanced Driver Assistance Systems
AFX	Animation Framework eXtention
API	Application Programming Interface
APM	Augmented Printed Material
AR	Augmented Reality
ARAF	Augmented Reality Applications Format
ARF	Augmented Reality Framework
ARML	Augmented Reality Markup Language
ASCII	American Standard Code for Information Interchange
ASTM	American Society for Testing and Materials
AUI	Advanced User Interface
AVC	Advanced Video Coding
BBA	Bone Based Animation
BIFS	Binary Format for Scenes
bpv	bits per vertex
CAD	Computer Aided Design
CAVE	Cave Automatic Virtual Environment
CCOM	Common Collaborative Object Model
CDVS	Compact Descriptors for Visual Search

CI	Coordinate Interpolator
CIF	Common Industry Format
CMMS	Computerized Maintenance Management System
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DAE	Digital Asset Exchange
DB	Data Base
DIN	Deutsches Institut für Normung
DOM	Document Object Model
EAM	Enterprise Asset Management
ERP	Enterprise Resource Planning
FAMC	Frame-based Animated Mesh Compression
FBA	Face and Body Animation
FBX	FilmBoX - data format
GIS	Geographic Information System
glTF	GL Transmission Format
GML	Geography Markup Language
GPS	Global Positioning System
GR	Group Report
HEVC	High Efficiency Video Coding
HMD	Head Mounted Display
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IFC	Industry Foundation Classes
IP	Internet Protocol
ISO	International Organization for Standardization
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
JPEG	Joint Photographic Experts Group
JSON	JavaScript Object Notation
KARML	Keyhole Augmented Reality Application Format
KML	Keyhole Markup Language
LOD	Level of Details
LUT	Lookup Table
MAR	Mixed and Augmented Reality
MAR-RM	Mixed and Augmented Reality Reference Model
MPEG	Motion Picture Expert Group
OGC	Open Geographic Consortium
OI	Orientation Interpolation
OPC	Open Plateforme Communication
PAS	Publicly Available Specification
PCC	Point Cloud Compression
PFS	Progressive Forest Split
PI	Position Interpolation
PM	Progression Mesh
PNG	Portable Network Graphics
PROTO	VRML Prototype
QR	Quick Response
RFC	Requests For Comments
RTCP	Real Time Control Protocol
RTP	Real Time Protocol
RTSP	Real Time Streaming Protocol
SCTP	Stream Control Transmission Protocol
SDK	Software Development Kit
SLAM	Synchronous Location and Mapping
SPEC	Specification
SVG	Scalable Vector Graphics
TCP	Transmission Control Protocol
TFAN	Triangle Fan
TS	Topological Surgery

UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VR	Virtual Reality
VRML	Virtual Reality Markup Language
WFS	Web Feature Service
WMS	Web Map Service
WSS	Wavelet Subdivision Surface
XML	eXtensible Markup Language
XR	eXtended Reality

4 Standards for augmented reality

4.1 Introduction

This clause describes the existing technical specifications published by various SDOs and directly addressing AR applications and services. At the time of publishing the present document, the following SDOs producing AR specific standards were identified:

- International Organization for Standardization (ISO/IEC) JTC1 https://www.iso.org/isoiec-jtc-1.html.
- Open Geospatial Consortium (OGC) http://www.opengeospatial.org/.
- World Wide Web Consortium (W3C) https://www.w3.org/.

Table 1 indicates the AR related standards produced by these organizations.

Table 1: AR related standards

ISO/IEC JTC1	ARAF - Augmented Reality Application Format	ISO IEC 23000-13 [i.2]	https://www.iso.org/standard/69465.html
	MAR-RM Mixed and Augmented Reality Reference Model	ISO/IEC 18039 [i.1]	https://www.iso.org/standard/30824.html
OGC	ARML Augmented Reality Markup Language [i.2]		https://www.opengeospatial.org/standards/arml
W3C	WebXR [i.2]		https://www.w3.org/blog/tags/webxr/

In the following clauses, each of these standards are briefly introduced.

4.2 Augmented Reality Application Format (ARAF)

Augmented Reality Application Format (ISO/IEC 23000-13 [i.2]) focuses on the data format used to provide an augmented reality presentation and not on the client or server procedures. ARAF specifies scene description elements for representing AR content, mechanisms to connect to local and remote sensors and actuators, mechanisms to integrate compressed media (image, audio, video, graphics), mechanisms to connect to remote resources such as maps and compressed media. ARAF was developed by MPEG, the same technical committee that created mp3 for audio, AVC and HEVC for video. MPEG already provided data type representations for all kinds of media, from static image, video, audio to 3D graphics and complex dynamic scenes. Additionally, MPEG developed a set of standards related to sensors and actuators. By bringing these two components together into an application format called ARAF, MPEG enables interoperability when used to build AR applications and services.

Table 2 synthesizes the set of existing MPEG tools that are used for addressing AR applications.

Table 2: MPEG tools addressing AR applications

Feature	Standard
Image (video) & audio capturing	MPEG-V published as ISO/IEC 23005-5 [i.11]
Capture real camera position and orientation	MPEG-V published as ISO/IEC 23005-5 [i.11] CDVS published as ISO/IEC 15938-13 [i.37]
Detection and tracking of visual objects	CDVS published as ISO/IEC 15938-13 [i.37]
Representation and transmission of media assets	MPEG-4 Systems published as ISO/IEC 14496-1 [i.3] MPEG-4 Visual published as ISO/IEC 14496-2 [i.4] MPEG-4 Audio published as ISO/IEC 14496-3 [i.5] MPEG-4 AVC published as ISO/IEC 14496-10 [i.6] MPEG-4 BIFS published as ISO/IEC 14496-11 [i.7] MPEG-4 AFX published as ISO/IEC 14496-16 [i.8]
Image & video rendering as a background	MPEG-4 Systems published as ISO/IEC 14496-1 [i.3]
Control the virtual camera parameters	MPEG-4 BIFS published as ISO/IEC 14496-11 [i.7]
Synthetic content representation	MPEG-4 BIFS published as ISO/IEC 14496-11 [i.7] MPEG-4 AFX published as ISO/IEC 14496-16 [i.8]

Therefore, by using these MPEG technologies it is possible to envision a generic AR browser. Instead of downloading a new and heavy application for every context, the users could simply point to an URL. The ARAF browser retrieves a scenario from the Internet, starts the video acquisition, tracks images and objects, recognizes them from visual signatures, computes the camera position, downloads 2D/3D graphics, composes a new scene, gets input from various sensors and constantly adapts the interaction in order to offer an optimal ARAF experience. Instead of developing a new application for each use case and smart phone/glasses platform, the industry could rely on MPEG-compliant Authoring Tools and MPEG-compliant browsers to reach a maximum number of customers. The schema in figure 1 presents the MPEG vision on AR.

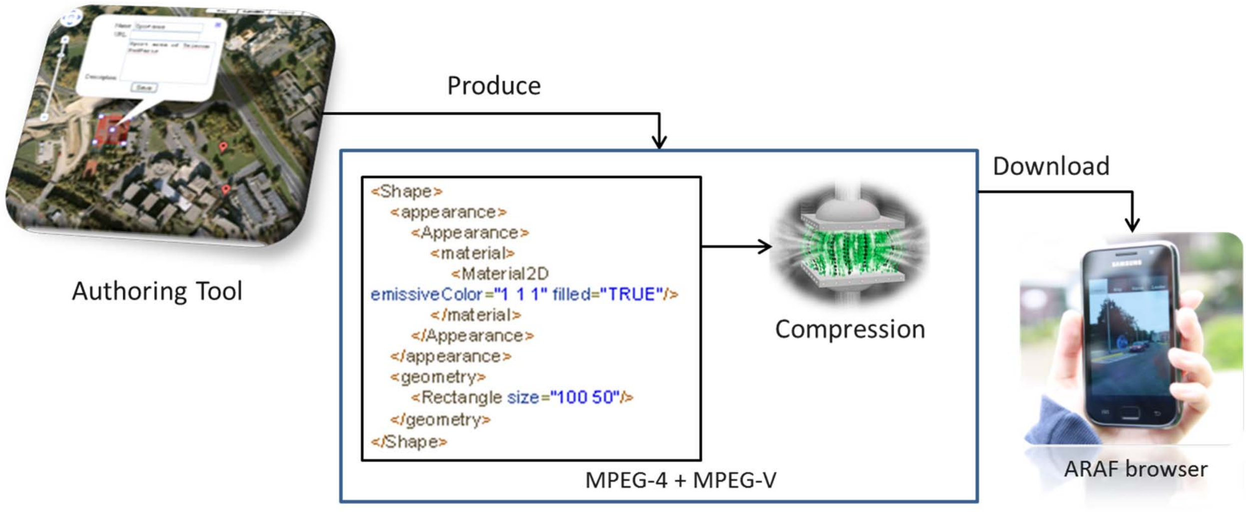

Figure 1: MPEG strategy on addressing AR [i.2]

The creation of ARAF content is fast and easy because the AR experience creator does not necessarily need to have any ARAF prior knowledge. Instead of designing the application from scratch, he can choose to create generic template-based AR applications in a matter of hours by using authoring tools. The authoring tool can be a user-friendly web interface where the user is allowed to design and define the behaviour of his AR experience. The generated result, an XML representation, and the media which is linked into the application, are compressed together in a single file which contains the full ARAF content. At the end of the chain, the content is downloaded and consumed by an ARAF browser.

On the other hand, the ARAF model provides full control to the designers who want to create brand new AR experiences. If one decides to create an AR experience from scratch, one can simply start writing the ARAF content by oneself, as one would create an HTML page.

Some key elements differentiating ARAF from other available technologies are the following:

1) it can be easily used by designers without a programming background;

2) it is an open standard provided by ISO; and

3) the software implementation of ARAF browser is available as open source.

Figure 2 presents a generic AR architecture including ARAF. The ARAF content, available as a file or stream, is interpreted by a device, called ARAF device. The elements (nodes) of the ARAF scene point to different sources of multimedia content such as 2D/3D image, audio, video and graphics, and sensor/sensory information sources/sinks that are either local or remote.

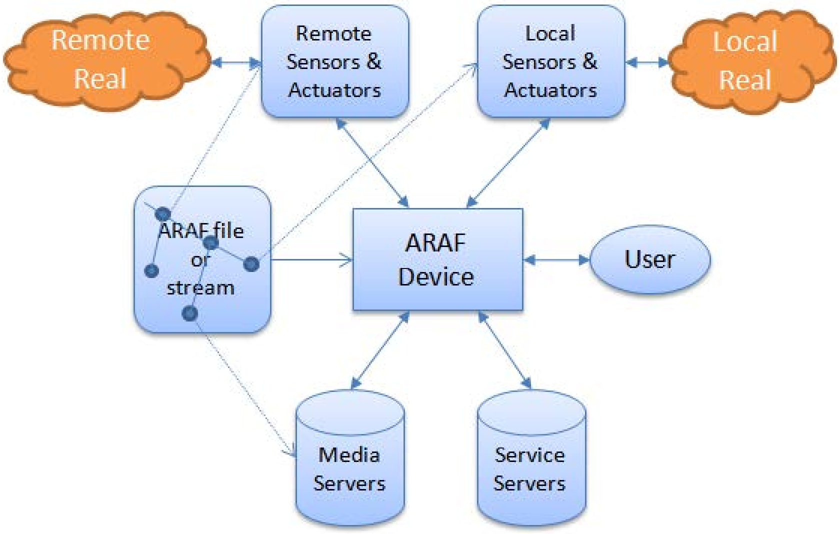

Figure 2: The ARAF context

In order to design a multimedia scene ARAF extends MPEG-4 Part 11 BIFS (ISO/IEC 14496-11 [i.7]) which is based on VRML97 (ISO/IEC 14772-1 [i.9]). About two hundred elements are standardized in MPEG-4 BIFS (ISO/IEC 14496-11 [i.7]) and VRML, giving the possibility of a content creator to create any AR experience that he can imagine.

MPEG-4 Part 11 describes a scene with a hierarchical structure that can be represented as a graph. The nodes (elements) of the graph build up various types of objects, such as audio, video, image, graphics and text. Furthermore, to ensure flexibility, a new user-defined type of node can be defined on demand by using the prototyping (PROTO) method.

In general, the nodes expose a set of parameters, through which aspects of their appearance and behaviour can be controlled. By setting these values, the content creators (designers) have the possibility to force a scene reconstruction or scene update at clients' terminals to adhere to their intention in a predefined manner. In more advanced scenarios, the structure of the BIFS nodes is not necessarily static; nodes can be added or removed from the scene graph arbitrarily and dynamically.

Certain types of nodes called sensors can interact with users and trigger appropriate actions, which are transmitted to other nodes through a routing mechanism, causing changes in the state of these receiving nodes. They are one of the bases for dynamic behaviour of multimedia content supported by MPEG-4 Part 11 (ISO/IEC 14496-11 [i.7]).

The maximum flexibility in the programmable feature of MPEG-4 scene is carried out with the Script node. By using the routing mechanism to a Script node, an associated function can be called. The implementation of this function is designed by the MAR experience creator, i.e. the creator can freely process some computations, and then output the result to the scene graph. In other words, by using the scripting feature of ARAF, the MAR experience creator can achieve a set of functionalities which are not supported natively by the nodes that are already part of the technology. This way the scene can be programmatically updated based on time events, user actions, or a mixture of the two. It is therefore provided the possibility of manipulating everything that is defined in the scene graph, at any moment in time and based on events that are triggered automatically or by the end user.

ARAF supports the definition and reusability of complex elements by using the MPEG-4 PROTO mechanism. The PROTO statement creates new nodes by defining a configurable object prototype; it can integrate any other node from the scene graph and it basically provides a method of creating new functionality that can be re-used within the scene graph.

Furthermore, a new functionality (a PROTO) that was designed by a MAR experience creator can be re-used in different scene graphs by using the EXTERNPROTO mechanism. This feature allows an object prototype to be described in a separate file and imported from there into any other scene graph.

The data that is captured from sensors or used to command actuators in ARAF is based on ISO/IEC 23005-5 [i.11], data formats for interaction devices (MPEG-V published in ISO/IEC 23005-5 [i.11]). MPEG-V provides architecture and specifies associated information representations to enable interoperability between virtual worlds. Concerning mixed and augmented reality, MPEG-V specifies the interaction between virtual and real worlds by implementing support for accessing different input/output devices, e.g. sensors, actuators, vision and rendering and robotics.

ARAF is capable of connecting to all sensors within the general set of sensors provided by smart phones today:

- orientation;
- position;
- acceleration;
- angular velocity;
- GPS;
- altitude;
- geomagnetic;
- camera; and
- microphone.

In addition to reading data from sensors, ARAF can also transmit data to actuators, controlling therefore physical devices directly from the scene graph. Here is the list of actuators that are supported in ARAF: light, vibration, tactile, flash, heating, cooling, wind, sprayer, scent, fog, rigid body motion, and kinaesthetic.

Table 3 specifies the AR features supported by ARAF.

Table 3: AR features supported by ARAF

Elementary media	Audio
	Image and video
	Textual information
	Graphics
Programming information	
User interactivity	Time sensors
	Touch sensors
	Media sensors
	Access to sensors and actuators of physical devices
Scene graph related information	Organization of scene elements
	Navigation information
	Layouting
	Visual identification and tracking, local and remote
	Remote visual registration and composition
	Audio identification and synchronization, local and remote
User localization	GPS
Map support	
Dynamic and animated scene	Interpolators and valuators
	Scripting
	Sensors
Communication and compression	Media control
	Map support
Terminal capabilities	

The ARAF content is encoded in textual BIFS, XML or in the equivalent binary representation encoded by BIFS. Figure 3 shows a short fragment from an ARAF example encoding the Augmented Printed Material (APM) use case. APM is an AR application that enriches (physical) printed material with any digital media-like videos, images, sounds or 3D graphics. The application presents the user additional information related to the printed material that he or she is reading. An APM application can enrich anything from a simple book to a tourism guide, a city map or a newspaper.

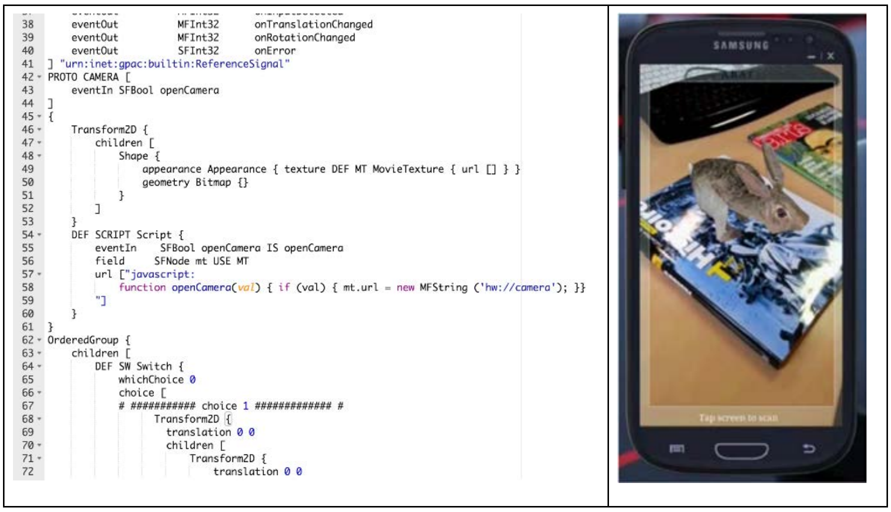

Figure 3: A fragment from the ARAF file for APM use-cases illustrated on right side of the figure

4.3 Augmented Reality Markup Language (ARML 2.0)

"Augmented Reality Markup Language 2.0 (ARML 2.0)" [i.49], published by OGC, is a descriptive XML-based data format. Initially, ARML 1.0 was a working document extending a subset of KML (Keyhole Mark-up Language) to allow richer augmentation for location-based AR services. While ARML uses only a subset of KML, KARML (Keyhole Augmented Reality Mark-up Language) uses the complete KML format. KARML tried to extend even more KML, offering more control over the visualization. By adding new AR-related elements, KARML deviated a lot from the original KML specifications. ARML 2.0 combined features from ARML 1.0 and KARML and has been released as an official OGC Candidate Standard in 2012 and approved as a public standard in 2015. While ARML 2.0 does not explicitly rule out audio or haptic AR, its defined purpose is to deal only with mobile visual AR.

ARML is built on a generic object model that allows serialization in several languages. Currently, ARML defines an XML serialization, as well as a JSON serialization for the ECMAScript bindings. The ARML object model consists of three main concepts:

1) Features - represent the physical object that should be augmented;

2) VisualAssets - describe the appearance of the virtual object in the augmented scene; and

3) Anchors - describe the spatial relation between the physical and the virtual object.

As a requirement, a device running an AR implementation using ARML 2.0 has a component (screen, see-through display, etc.) where the virtual objects are projected on. The device has sensors such as a camera, GPS, and orientation to analyse the real world.

Users interact with the virtual scene by moving around in the real world. Based on the movement of the user, the scene on the screen is constantly updated. A user can also interact with the scene by selecting virtual objects, typically by touching them on the screen. However, how a user can select a virtual object is application- and device-specific and out of scope for ARML 2.0 [i.49].

As shown in figure 4, in ARML 2.0 [i.49], a Feature represents a real world object that should be augmented. Technically speaking, a Feature consists of some metadata on the real world object, as well as one or more Augmentations that describe where a Feature is located in the composed scene. In ARML 2.0 terms, an Augmentation is called an Anchor. Anchors define the link between the digital and the physical world (a broader concept of a *location*). An Anchor describes where a particular Feature is located in the real world. An Anchor can be either a spatial location that is tracked using location and motion sensors on the device, or a visual pattern (such as markers, QR codes or any sort of reference image) that can be detected and tracked in the camera stream using computer vision technology. Finally, VisualAssets describe how a particular Anchor should be represented in the Composed Scene. VisualAssets can either be 2-dimensional (such as text or images) or 3-dimensional.

```
<arml xmlns="http://opengis.net/arml/2.0"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:gml="http://www.opengis.net/gml/3.2">
    <ARElements>
        <Feature id="myFeature1">
            <name>HFT Building1</name>
                <anchors>
                    <gml:Point>
                        <gml:pos>48.780570 9.173404</gml:pos>
                    </gml:Point>
                    <picture>Picture of HFT Building 1</picture>
                    <video>Main Entrance_2.mp4</video>
                </anchors>
        </Feature>
    </ARElements>
</arml>
```

Figure 4: An ARML example for geographic position augmentation with a video

ARML is expected to be extended in the future to support non-visual virtual objects, such as sound and haptic feedback. The current specification of ARML 2.0 [i.49], however, focuses on visual objects. ARML 2.0 is built on top of a generic object model to allow future serializations in different languages, as well as good extensibility for future needs. Figure 5 shows the generic object model.

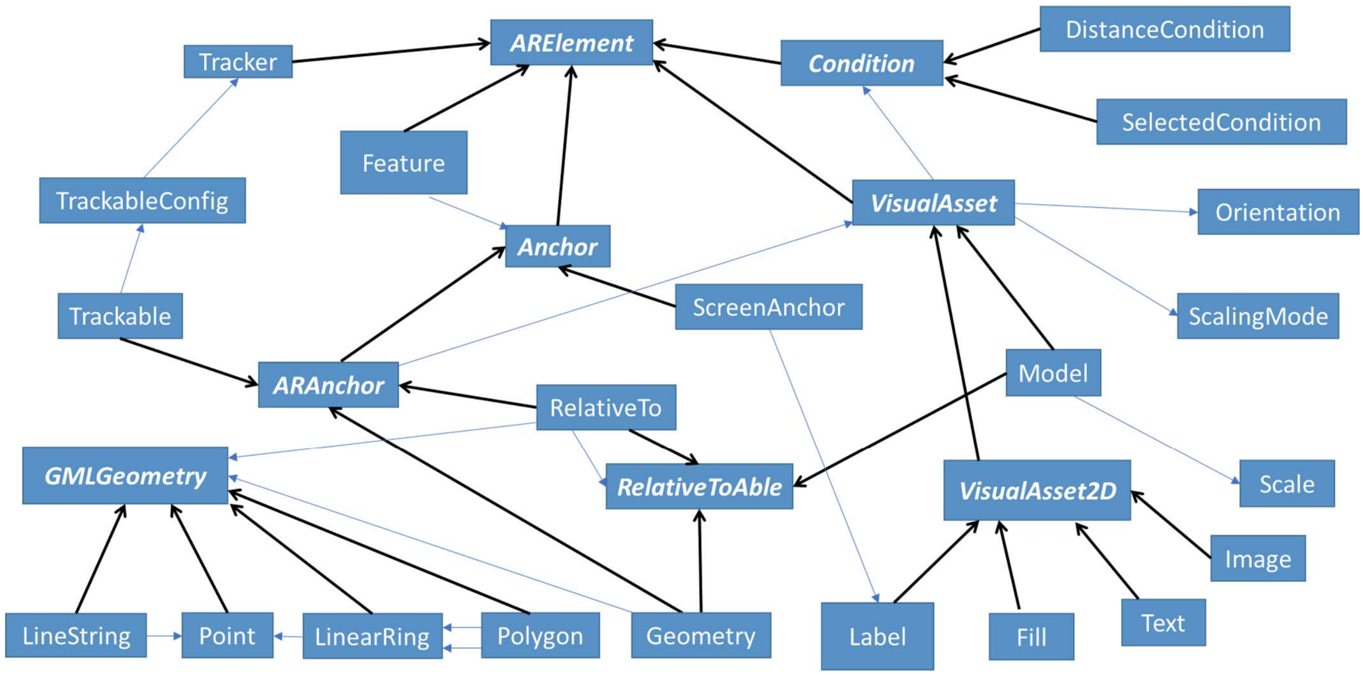

Figure 5: Generic object model

4.4 W3C WebXR

The WebXR [i.55] Device API provides access to input and output capabilities commonly associated with Virtual Reality (VR) and Augmented Reality (AR) hardware. By using this API, it is possible to create Virtual Reality and Augmented Reality web sites that can be viewed with the appropriate hardware like a VR headset or AR-enabled phone.

1) the goals of the API are to enable XR applications on the web by allowing pages to do the following:to detect if XR capabilities are available;
2) to query the XR devices capabilities;
3) to poll the XR device and associated input device state; and
4) to display imagery on the XR device at the appropriate frame rate.

Use cases can be games, but also 360 and 3D videos and object and data visualization.

The basic steps most WebXR applications will go through are the following:

1) Request an XR device.
2) If a device is available, application advertises XR functionality to the user (for example, by adding a button to the page that the user can click to start XR content).
3) Request an exclusive XR session from the device in response to a user-activation event.
4) Use the session to run a render loop that produces graphical frames to be displayed on the XR device.
5) Continue producing frames until the user indicates that they wish to exit XR mode.
6) End the XR session.

Web XR [i.55] specification then takes into consideration the following items:

- it describes security, privacy and comfort elements to be considered: to prevent for example risk of malicious page to infer what a user is typing, for compositing the trusted and untrusted content or about the isolation of the context of pages;

- it permits to enumerate the list of devices attached to the system and describes how to manage it (list, identify default device, identify if the device can support immersive sessions and how);
- it manages XR-sessions: creation (with session description), initialization and shut down. Each XR session has an environment-blending-mode value (for example opaque for virtual reality devices, additive or alpha-blend for Augmented Reality devices);
- it manages the animation of frames: request, cancel and run the animation frame;
- it handles the presentation to the XR Device;
- it describes the XR view into an XR Scene which corresponds to a display used by the XR device to present imagery to the user. Many views can be requested (one for each eye for example). The XR viewport is also described as a rectangular region of a graphics surface;
- it deals with pose by providing various transforms in the form of matrices and describing the position and orientation of the XR Device;
- it deals with actions (like pressing a trigger, touchpad or button);
- it describes the source of images and the compatibility with WebGL context.

At the time of publication of the present document, the specification is still under development.

4.5 Mixed and Augmented Reality Reference Model (MAR-RM)

This International Standard, published as ISO/IEC 18039 [i.1], is a technical report defining the scope and key concepts of mixed and augmented reality, the relevant terms and their definitions, and a generalized system architecture that together serve as a reference model for Mixed and Augmented Reality (MAR) applications, components, systems, services, and specifications. This reference model establishes the set of required modules and their minimum functions, the associated information content, and the information models that have to be provided and/or supported to claim compliance with MAR systems.

The reference model is intended for use by current and future developers of MAR applications, components, systems, services, or specifications to describe, compare, contrast, and communicate their architectural design and implementation. The MAR-RM is designed to apply to MAR systems independent of specific algorithms, implementation methods, computational platforms, display systems, and sensors or devices used.

This International Standard does not specify how a particular MAR application, component, system, service, or specification could be designed, developed, or implemented. It also does not specify the bindings of those designs and concepts to programming languages, or the encoding of MAR information through any coding technique or interchange format. This specification contains a list of representative system classes and use cases with respect to its reference model.

5 Standards for AR-related data representation

5.1 Introduction

In AR applications, two kinds of data are generally mixed: a digital representation of the real world captured by different sensors and natural or computer generated visual and aural assets under various forms such as text, images, videos, graphics objects, audio elements. Several SDOs specified data formats for this different kind of contents and this clause briefly introduces them. At the time of publishing the present document the following SDOs and consortia were identified:

- International Organization for Standardization (ISO/IEC) JTC1 https://www.iso.org/isoiec-jtc-1.html for multi-media data.
- World Wide Web Consortium (W3C) https://www.w3.org/ for multi-media data.
- Open Geospatial Consortium (OGC) http://www.opengeospatial.org/ for geographical data.

- Internet Engineering Task Force (IETF) https://www.ietf.org for geographic data.

5.2 Text

Using text is one of the simplest manner to augment a scene. There are various ways to express textual data, from simple ASCII strings to more or less complexed structures such as JSON (published by IETF under the number IETF RFC 7159 [i.41]), XML [i.56] or HTML [i.57] (both published by W3C) and PDF (published by ISO with the number ISO 19005-3 [i.12] Portable Document Format).

5.3 Image

There are several standards for representing images and the most used in AR applications are JPEG (published by ISO under the number ISO/IEC 10918 [i.13] and by Recommendation ITU-T T.81 [i.76]), PNG - Portable Network Graphics (published by ISO under the number ISO/IEC 15948 [i.14]) and JP2 - JPEG 2000 (published by ISO under the number ISO/IEC 15444-2 [i.15]).

5.4 Video

5.4.1 MPEG Video standards

MPEG developed several generations of video compression standards starting from MPEG-1 (ISO/IEC 11172-3 [i.18]) published in 1993 to MPEG-I (ISO/IEC 23090 [i.17]), a standard under development at the time of this report. The most used video standards in AR applications are MPEG-4 AVC Advanced Video Coding (co-published by ISO and ITU under the numbers ISO/IEC 14496-10 [i.6] and Recommendation ITU-T H.264 [i.39] respectively) and MPEG-H HEVC High Efficiency Video Coding, (co-published by ISO and ITU under the numbers ISO/IEC 23008-2 [i.16] and Recommendation ITU-T H.265 [i.40] respectively). Current standardization efforts are conducted in considering video 360° by initiatives such as MPEG-I (ISO/IEC 23090 [i.17]).

5.4.2 HTML5 Video Element

The HTML5 Video Element allows for playing and displaying video content. The source can be media files accessible via a URL or streams resulting from using the HTML5 Media Capture and Streams as well as the WebRTC recommendation [i.58]. The displaying browser provides the users with functionalities for starting, pausing, and stopping a playback.

5.5 Audio

5.5.1 MPEG Audio standards

MPEG developed several generations of audio compression standards starting from the well-known mp3 component of MPEG-1 published in 1993 to MPEG-I, a standard under development at the time of this report. The most used audio standards in AR applications are MPEG-1 mp3 (published by ISO under the numbers ISO/IEC 11172-3 [i.18]), MPEG-4 AAC Advanced Audio Coding (published by ISO under the number ISO/IEC 14496-3 [i.5]) and MPEG-D Spatial Audio Coding (ISO/IEC 23003 [i.19]). Current standardization efforts are conducted in considering immersive audio by initiatives such as MPEG-I (ISO/IEC 23090 [i.17]).

5.5.2 HTML5 Audio Element

The HTML5 Audio Element allows for playing and rendering audio content. The source can be media files accessible via a URL or streams resulting from using the HTML5 Media Capture and Streams as well as the WebRTC recommendation [i.58]. The displaying browser provides the users with functionalities for starting, pausing, and stopping a playback.

5.6 2D graphics objects

5.6.1 Simple Vector Graphics

The W3C Scalable Vector Graphics (SVG) [i.59] is an XML based specification for describing graphics. The nature of the graphics is vector based and allows for scalability. This is of interest for AR applications, as different resolutions are required for different devices, display sizes and situations in which graphics need to be displayed. In addition, it is possible to filter for parts of graphics, i.e. layers to display at a certain time. Through this, also the management of transparencies is possible.

5.7 3D graphics objects and scenes

5.7.1 Non-compressed data formats

In the last decade, several efforts have been made to develop a unique data format for interchanging assets between 3D graphics authoring tools e.g. open standards, X3D [i.10] (based on VRML [i.9]), and COLLADA [i.62]. While COLLADA concentrates on representing 3D objects or scenes, X3D pushes the standardization further by addressing as well user interaction thanks to an event model in which scripts, possibly external to the file containing the 3D scene, may be used to control the behaviour of its objects. One of the characteristics of both X3D and COLLADA is the data representation based on XML. Indeed, using XML allows a very easy extension of the structure, as one can impregnate data with meaning and purpose, implicitly upgrading it to be self-contained information. XML is an appropriate mechanism to express dynamic 3D scenes because even non-interactive ones usually contain complex structures of heterogeneous data, organized in different kinds of graphs (for both the scene as a whole and its possibly many objects) and containing elements of inherently different nature (geometry, appearance, animation, lights, viewpoints, etc.).

Another recent format for 3D graphics assets is glTF [i.63]. It is designed for efficient transmission and loading of 3D scenes and models, however is not including a real compression method. It minimizes both the size of 3D assets, and the runtime processing needed to unpack and use those assets. glTF defines an extensible, common publishing format for 3D content tools and services that streamlines authoring workflows and enables interoperable use of content across the industry.

5.7.2 Compressed data formats

Generic compression on top of XML of JSON (achieved through entropy coding of the corresponding text file thanks to "gzip" or similar tools) usually reduces the data size only by a factor 10, because it is just able to exploit the data structure redundancy, since the information represented by the data is not understood and hence cannot be exploited. Even worse, streaming and scalable coding are out of the question with generic compression methods since both require that the data semantics be understood to build hierarchical or at least progressive representations of the information, which can then be adequately packetized. Instead, specific 3D graphics (lossy) compression techniques may exploit better the spatial or temporal correlation present in the information and reduce the data size by a factor of over 40 while possibly yielding progressive, scalable bitstreams, suitable for streaming scenarios involving heterogeneous networks and/or terminals.

Built on top of VRML, MPEG-4 contained, already in its first two versions published in 1999, tools for the compression and streaming of 3D graphics assets, enabling to describe compactly the geometry and appearance of generic, but static objects, and also the animation of human-like characters. Since then, MPEG has kept working on improving its 3D graphics compression toolset and published two editions of MPEG-4 Part 16, AFX (Animation Framework eXtension) ISO/IEC 14496-16 [i.8], which addresses the requirements above within a unified and generic framework and provides many more tools to compress more efficiently more generic textured, animated 3D objects.

Table 4 lists those MPEG standards that address the compression of 3D assets.

Table 4: Compression of 3D assets by MPEG standards

Tool name	Data type to be compressed	Official standard name
3DMC (3D Mesh Coding)	Shape	ISO/IEC 14496-2 [i.4] Visual
FBA (Face and Body Animation)	Avatar animation	ISO/IEC 14496-2 [i.4] Visual
3DMCe (3D Mesh Coding extension)	Shape	ISO/IEC 14496-16 [i.8] AFX
WSS (Wavelet Subdivision Surface)	Shape	ISO/IEC 14496-16 [i.8] AFX
TFAN (Triangle Fan), part of SC3DMC (Scalable Complexity 3D Mesh Coding)	Shape	ISO/IEC 14496-16 [i.8] AFX
CI, OI, PI (Coordinate, Orientation and Position Interpolators)	Animation	ISO/IEC 14496-16 [i.8] AFX
BBA (Bone-Based Animation)	Skeleton animation	ISO/IEC 14496-16 [i.8] AFX
FAMC (Frame-based Animated Mesh Compression)	Dynamic mesh	ISO/IEC 14496-16 [i.8] AFX
PCC (Point Cloud Compression)	Static and dynamic point clouds	ISO/IEC 23090-5 [i.20] and ISO/IEC 23090-9 [i.21] (ongoing)

In the following paragraphs these compression methods are briefly introduced.

As in the cases of image or video coding, the concepts of resolution scalability and of *single- vs. multi-rate coding* are important for 3D mesh coding. In some scenarios, a single-rate coding of a 3D mesh may be enough, but its progressive compression may be desirable, especially if it is a complex mesh to be transmitted over a network with a restricted bandwidth, or to terminals with limited processing power. In such cases, it is useful to represent and code the original, fine mesh as a sequence (or a hierarchy) of refinements applied to a simple, coarse mesh. During decoding, connectivity and geometry are reconstructed incrementally from the bitstream until the original mesh is rendered at its full resolution or the transmission is cancelled by the user. Progressive compression thus allows transmission and rendering of different Levels Of Details (LODs).

Topological Surgery (TS) is a scheme for 3D mesh connectivity coding, which starts by cutting the mesh along a selected set of edges to create a vertex tree and a triangle tree. The mesh connectivity is then represented by both trees, yielding 1 bpv (bit per vertex) for very regular meshes and 4 bpv on average otherwise - vs. the raw size estimate of 96 bpv. TS was the basis for the 3D Mesh Coding (3DMC) toolset of MPEG-4 Part 2 (ISO/IEC 14496-2 [i.4]), and also for the 3DMC extension (3DMCe) tool of AFX, which adds support for efficient texture coordinate compression, and for mesh animation/editing.

As for geometry information, to achieve higher compression rates than the 3-16 bpv mentioned above, both 3DMC and 3DMCe feed the vertex coordinates to a *quantization* step, whose resulting values are then compressed by entropy coding after some *prediction* (relying on some smoothness assumptions) is applied. Both steps contribute to the compactness of the final result, but quantization is intrinsically and irreversibly lossy, whereas prediction is a perfectly reversible and lossless transformation of the signal to make it fit for a more efficient subsequent entropy coding. Vertex coordinates are typically uniformly quantized with 8-14 bits each and prediction is usually linear: at most three vertices adjacent to the one being decoded, and already decoded themselves, according to the vertex ordering imposed by TS, are used for the prediction known as the "parallelogram rule". This yields bitrates of some 13-18 bpv for 9-12 bits per coordinate, or 13 bpv at 8-bit quantization resolution.

The Scalable Complexity 3D Mesh Compression (SC3DMC) toolset of AFX specifies a way to fine tune the trade-off between compression efficiency (bpv) and computational resources (CPU/GPU and memory) needed in both encoder and decoder by choosing among three 3D mesh coding techniques: Quantization-Based Compact Representation, Shared Vertex Analysis and Triangle fan. The main idea behind the SC3DMC scheme is that, in some application scenarios, especially the ones involving mobile devices with reasonable network connections, the minimization of bitstream size may not be as important as that of computational resources. Current implementations of SC3DMC yield encoding performances of 17-70 bpv (for both topology and geometry) with associated decoding speeds of millions of vertices per second on an ordinary computer.

The concept of Progressive Mesh (PM) allows to code a mesh with a total of ~35 bpv. A PM is a base mesh plus a sequence of vertex split records, each specifying which vertex and pair of edges incident to it should be split, and the local geometry changes. From such a representation, it is easy to extract an LOD of the mesh with any desired number of triangles by simply choosing the adequate prefix of the vertex split sequence, which is streamed after the base mesh has been transmitted. The Progressive Forest Split (PFS) technique is based on the PM and TS ideas and was included in the old 3DMC toolset of MPEG-4 Part 2 (ISO/IEC 14496-2 [i.4]). PFS is able to reduce the bitrates of PMs at the expense of reduced granularity: two successive LODs of a PFS set differ by a group of vertex splits, instead of only one.

Logically enough, the highest compression ratios are achieved by minimizing the number of LODs but, typically, it is possible to remain slightly below 30 bpv for medium size meshes coded with several LODs.

Wavelet Subdivision Surfaces (WSSs) are the basis for the best known truly *hierarchical* (as opposed to merely progressive) 3D mesh coding technique. The connectivity information can be coded extremely efficiently thanks to remeshing. Besides, the geometry information contained in the hierarchical set of 3D details (representing the prediction errors between successive LODs) can also be very compactly coded as follows: *i)* a space-frequency wavelet transform is applied to the 3D details to obtain a zero-centred set of coefficients; *ii)* since the magnitude of those wavelet coefficients decays at finer levels with a rate related to the smoothness of the original surface, they are particularly well suited for zero tree coding. In fact, the coding efficiency results of the WSS tool in AFX are over four times better than those of PFS (in terms of reconstruction error for a given bitrate).

Key frame animation (and implicitly the corresponding value interpolation between key frames) is one of the most popular methods for computer animation. The interpolated data type determines the dimension of the value in each (key; value) pair: for vertex coordinates and object positions, 3D values have to be coded/interpolated, whereas orientations are represented with 4D vectors called quaternions (an axis plus an angle). The old Interpolator Compression tool of MPEG-4 Part 11 (ISO/IEC 14496-11 [i.7]) was designed to compress generic interpolated data and can achieve compression ratios of up to 30:1 (vs. textual representation of the same data) for coordinates, normal, translations and rotations. However, the Interpolator Compression tool is unable to exploit any spatial redundancy present in the animation data.

AFX's Frame-based Animated Mesh Compression (FAMC) tool does cluster vertices based on their motion properties. FAMC encodes the time-varying positions, normal, etc. associated to the vertices of a mesh, both for deformations and rigid motions. The data in a FAMC stream is structured into segments of several frames that can be decoded individually. Within a segment, a temporal prediction model, used for motion compensation, is represented. Each decoded animation frame updates the geometry and possibly the attributes of the 3D object that FAMC is referred to. Once the mesh is segmented with respect to the motion of each cluster, three kinds of data are obtained and encoded. The standardized implementation of the FAMC encoder achieves compression ratios of up to 45:1 (vs. textual representation of the same data).

Concerning avatar animation, one technique is the Face and Body Animation (FBA) tool of MPEG-4 Part 2 (ISO/IEC 14496-2 [i.4]), which was highly inspired on video coding algorithms, and targeted at the compression of virtual humanoids at very low bit-rate, had limitations in terms of realism.

AFX includes, since its first edition, a more powerful and generic tool, called Bone-Based Animation (BBA), featuring a multilayer modelling of the avatar and mesh deformation driven by a set of 1D controllers (bones and muscles). BBA is a compression tool for geometric transforms of bones (used in skinning-based animation) and weights (used in morphing animation). These elements are defined in the scene graph and should be uniquely identified so that the BBA stream may then refer to them. The BBA stream contains, for each frame, the new transforms of bones (expressed as Euler angles or quaternions) and the new weights of the morph targets.

Analogously to what was done in the FBA framework, two compression schemes were developed in BBA to encode the Skeleton, Muscles and Skin animation parameters, namely predictive-based and DCT-based. The compression ratios obtained with both encoding schemes depend strongly on the motion and skeleton complexity, and on the number of muscles used. Typically, when considering a human-like virtual character with natural motion (obtained from a motion capture system), a bit rate of 25 kb/s ensures a good motion quality, although some optimization mechanisms allow achieving better results. Without an a priori knowledge about the range of bone and muscle motion, due to the fact that any kind of skeleton is allowed, the MPEG-4 standard cannot directly address an efficient arithmetic coding. For this reason, the standard supports the transmission of this range within the animation stream. Optimized implementation of the BBA encoder yields compression ratios of up to 70:1 (vs. textual representation of the same data).

Advanced 3D representations of the real world are enabling more immersive forms of interaction and communication to better understand and navigate it. 3D point clouds have recently emerged as such an enabling representation. Clouds of 3D points with associated attributes (colour, material properties, etc.) can also be generated synthetically, but are typically captured using multiple cameras and depth sensors in various setups and may have to contain from thousands up to billions of points in order to realistically represent the sampled objects or scenes, which can later be reconstructed/rendered thanks to those point clouds. Point Cloud Compression (PCC) technologies are thus obviously needed: lossy PCC is likely to be desirable for use in real-time communications, and lossless PCC might be necessary in the contexts of dynamic mapping for autonomous driving, six Degrees of Freedom (6 DoF) virtual reality, cultural heritage applications, etc. The MPEG Video-based Point Cloud Compression (V-PCC) technology is leveraging existing and future video codecs and video eco-systems in general (hardware acceleration, transmission services and infrastructure) while enabling new kinds of applications. It addresses lossless and lossy coding of 3D point clouds with associated attributes such as colour. The current V-PCC test model encoder implementation shows compression performances of 125:1 while achieving good perceptual quality. Point clouds are typically represented by extremely large amounts of data, which is a significant barrier for mass market applications. They are becoming increasingly popular to present immersive volumetric video due to the relative ease of capture and render when compared to other volumetric video representation. Several applications include six Degrees of Freedom (6 DoF) immersive media, VR/AR, immersive real-time communication, autonomous driving, cultural heritage and a mix of individual point cloud objects with background 2D/360 video. V-PCC is considered to provide high-level immersiveness at currently available bandwidths for deployment of said applications on future networks.

5.8 Scene description

5.8.1 VRML

The ISO/IEC 14772-1 [i.9] (also known as Virtual Reality Modeling Language) defines a file format that integrates 3D graphics and multimedia. Conceptually, each VRML file is a 3D time-based space that contains graphic and aural objects that can be dynamically modified through a variety of mechanisms. This part of ISO/IEC 14772-1 [i.9] defines the interface that applications external to the VRML browser may use to access and manipulate the objects defined in ISO/IEC 14772-1 [i.9].

The interface described is designed to allow an external environment to access nodes in a VRML scene using the existing VRML event model. In this model, an eventOut of a given node can be routed to an eventIn of another node. When the eventOut generates an event, the eventIn is notified and its node processes that event. Additionally, if a script in a Script node has a reference to a given node it can send events directly to any eventIn of that node and it can read the last value sent from any of its eventOuts.

The scope of this part of ISO/IEC 14772-1 [i.9] is to cover all forms of access to a VRML browser from external applications. It is equally valid for a database with an object interface to access a standalone browser in a presentation slide as it is for a Java applet operating within a web browser and the available services do not vary.

This part of ISO/IEC 14772-1 [i.9] does not provide a byte level protocol description as there can be many valid ways of expressing an interaction with a browser. Instead, it represents the interface in terms of the services provided and the parameters that are passed to access these services. Individual language and protocol bindings to these services are available as annexes to this part of ISO/IEC 14772-1 [i.9].

5.8.2 X3D

ISO/IEC 19775 [i.10] (also known as X3D) is a royalty-free open standards file format and run-time architecture to represent and communicate 3D scenes in multiple applications. The X3D family of standards is ratified by the International Standards Organization (ISO) to ensure archival stability and steady evolution. X3D graphics provides a system for the storage, retrieval, and playback of real-time graphics content embedded in applications, all within an open architecture to support a wide array of domains and user scenarios.

5.8.3 MPEG-4 Part 11 BIFS (ISO/IEC 14496-16)

One of the biggest steps forward of MPEG-4 with respect to its predecessors, MPEG 1 and MPEG-2, was the definition of BIFS as a part of the system information, which lies on top of all media data and acts as an intermediate layer between media data and the final displayed content. It gives a flexible way to manipulate various types of audio-visual media in an MPEG-4 interactive scene, to schedule events, to coordinate multimedia objects both in the temporal and spatial domains, to process interactivity, etc. BIFS is based largely on VRML97: it is a very efficient, binary encoded version of an extended set of VRML97 nodes. Similarly to VRML, BIFS includes tools to handle the user input (sensors), propagate actions in the scene graph (routes) and specify application behaviour (scripts and Java applets). Thus it is possible to program complete 2D/3D games and interactive applications that are interpreted by a compliant player or browser.

5.9 Standards related to sensors and actuators data

5.9.1 MPEG-V

MPEG-V (Media context and control), published in ISO/IEC 23005-5 [i.11], provides an architecture and specifies associated information representations to enable the interoperability between virtual worlds, e.g. digital content providers of a virtual world, (serious) gaming, simulation, and with the real world, e.g. sensors, actuators, vision and rendering, robotics. MPEG-V is applicable in various business models/domains for which audiovisual contents can be associated with sensorial effects that need to be rendered on appropriate actuators and/or benefit from well-defined interaction with an associated virtual world.

The strong/well-defined connection (defined by an architecture that provides interoperability through standardization) between the virtual and the real world is needed to reach simultaneous reactions in both worlds to change in the environment and human actions. Efficient, effective, intuitive and entertaining interfaces between users and virtual worlds are of crucial importance for their wide acceptance and use. To improve the process of creating virtual worlds a better design methodology and better tools are indispensable.

The MPEG-V standard consists of different parts handling various aspects such as the architecture, the control and sensory information, the virtual world object characteristics, the data format for interaction devices as well as the conformance and the reference software.

5.9.2 W3C Media Capture and Streams

Media Capture and Streams [i.60] is a World Wide Web Consortium candidate recommendation for accessing local media streams on the device that renders a website. The streams can be the video of a web camera or the audio of a microphone. The browser displaying a web site is responsible for handling this access and for requesting the access from the user. The resulting streams can be displayed or played with the HTML5 audio and video elements. The Media Capture and Streams recommendation is important for AR applications, as it allows capturing the real world environment.

5.10 Standards related to geographic data

5.10.1 Geography Markup Language (GML)

OGC GML [i.50] serves as a modelling language for geographic systems as well as an open interchange format for geographic transactions on the Internet. GML is mainly used for geographical data interchange, for example by Web Feature Service (WFS). WFS is a standard interface that allow exchanging geographical features between servers or between clients and servers. WFS helps to query geographical features, whereas Web Map Service is used to query map images from portals.

GML contains a rich set of primitives which are used to build application specific schemas or application languages. These primitives include: Feature, Geometry, Coordinate reference system, Topology, Time, Dynamic feature, Coverage (including geographic images), Unit of measure, Directions, Observations, Map presentation styling rules.

It is possible to define community-specific application schemas that are specialized extensions of GML. Using application schemas, users can refer to roads, highways, and bridges instead of points, lines and polygons. There are specific extensions like CityGML or SensorML.

5.10.2 City Geography Markup Language (CityGML)

OGC CityGML [i.51] is data model and exchange format to store digital 3D models of cities and landscapes. It defines ways to describe most of the common 3D features and objects found in cities (such as buildings, roads, rivers, bridges, vegetation and city furniture) and the relationships between them. It also defines different standard levels of detail (LoDs) for the 3D objects. LoD 4 aims to represent building interior spaces.

CityGML is implemented as a GML application schema. The structure of a CityGML file is a hierarchy that ultimately reaches down to individual objects and their attributes. These objects have a geometry that is described using GML.

The types of objects stored in CityGML are grouped into different modules: Appearance (textures and materials for other types), Bridge (bridge-related structures, possibly split into parts), Building (the exterior and possibly the interior of buildings with individual surfaces that represent doors, windows, etc.), CityFurniture (benches, traffic lights, signs, etc.), CityObjectGroup (groups of objects of other types), Generics (other types that are not explicitly covered), LandUse (areas that reflect different land uses, such as urban, agricultural, etc.), Relief (the shape of the terrain), Transportation (roads, railways and squares), Tunnel (tunnels, possibly split into parts), Vegetation (areas with vegetation or individual trees), WaterBody (lakes, rivers, canals, etc.). Tt is also possible to extend this list with new classes and attributes by defining Application Domain Extension.

The structure of a CityGML file is a hierarchy that ultimately reaches down to individual objects and their attributes. These objects have a geometry that is described using GML. Another important implementation of CityGML is the one of 3D City DB, which stores CityGML in a database.

5.10.3 Indoor Geography Markup Language (IndoorGML)

IndoorGML [i.52], specifies an open data model and XML schema for indoor spatial information. It represents and allows for exchange of geo-information that is required to build and operate indoor navigation systems. The targeted applications are indoor robots, indoor localization, indoor m-Commerce, emergency control, etc. IndoorGML does not provide spaces geometry but it can refer to data described in other format like CityGML, KML or IFC.

IndoorGML transforms real spaces into cellular spaces linked in a navigable diagram. IndoorGML supports multi-layer representation. Indoor trips can be made according to different modes. There are escalators, lifts, stairs, walkable areas, etc. The ways to go from a point A to a point B are numerous and dependent of many parameters. A way accessible by a walking man could be unreachable by a wheelchair. One might consider intangible spaces like radio communication cells and rules to pass from one cell to another. The IndoorGML model gives the ability to see spaces from different points of view, each with specific navigation rules.

Indoor positioning is not in the scope of IndoorGML but it can be used for navigation purpose.

5.10.4 Keyhole Markup Language (KML)

OGC KML [i.53] is an XML language focused on geographic visualization, including annotation of maps and images. Geographic visualization includes not only the presentation of graphical data on the globe, but also the control of the user's navigation in the sense of where to go and where to look. KML became an OGC standard in 2015 and some functionalities are duplicated between KML and traditional OGC standards.

The main applications target by KML are mobile maps (2D) and earth browsers (3D). KML and GML complement each other's, but a conversion in either way would lead to unrecoverable loss. Whereas GML is a language to encode geographic content for any application, by describing a spectrum of application objects and their properties (e.g. bridges, roads, buoys, vehicles, etc.). KML can be used to render GML content, and GML content can be "styled" using KML for the purposes of presentation.

KML contains descriptions of place marks, images, polygons, 3D models, textual descriptions, etc. Each place always has a longitude and a latitude, tilt, heading, altitude, which together define a "camera view".

5.10.5 Web Map Service (WMS)

OGC WMS [i.54] provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. The response to the request is one or more map images (returned as JPEG, PNG, etc). The standard is used to get different information layers from different compatible maps providers to blend them in a single, more informative image.

The WMS standard defines three operations: GetCapabilities, GetMap and GetFeatureInfo. GetCapabilities is used to obtain service-level metadata, which is a machine-readable (and human-readable) description of the WMS's information content and acceptable request parameters. GetMap is to be used to obtain a map image whose geospatial and dimensional parameters are well defined. Finally, GetFeatureInfo provides information about particular features shown on a map.

5.10.6 W3C GeoLocation

GeoLocation API [i.61] is a standardized interface to be used to retrieve the geographical location information from a client-side device. The location accuracy depends of the best available location information source (global position systems, radio protocols, Mobile network location or IP address location). Web pages can use the Geolocation API directly if the web browser implements it. It is supported by most desktop and mobile operating systems and by most web browsers. The API returns 4 location properties; latitude and longitude (coordinates), altitude (height), and accuracy.

5.10.7 IETF RFC 5870

IETF RFC 5870 [i.46] published by Internet Engineering Task Force (IETF), defines the 'geo' URI scheme. It defines a physical location in 2D or 3D coordinate system in a compact, human-readable and protocol-independent way. The default coordinate reference system used is WGS-84. As example, by using this specification, the address "http://www.openstreetmap.org/?lat=37.786971&lon=-122.39967&zoom=40" is written "geo:37.786971,-122.399677".

5.10.8 IETF RFC 7946

GeoJSON (IETF RFC 7946 [i.47) is a format that describes simple geographical features; Point, Line, Polygon, Multi-Point, Multi-LineString, MultiPolygon and non-spatial attributes. Its formalism is based on the JavaScript™ Object Notation. It is simple and human-readable. GeoJSON is largely adopted by mapping and GIS software and is also supported by current mapping services.

6 Standards for communication protocols

6.1 Introduction

AR applications and services may need real time communication protocols for transmission of data such as sensors data, audio, video or graphics assets. There are several standards for handling such communication starting from traditional TCP/IP and UDP to more dedicated ones like RTSP and WebRTC.

6.2 TCP, UDP

The Transmission Control Protocol (TCP) (IETF RFC 793 [i.48]) is one of the main protocols of the Internet protocol suite. It originated in the initial network implementation in which it complemented the Internet Protocol (IP). Therefore, the entire suite is commonly referred to as TCP/IP. TCP provides reliable, ordered, and error-checked delivery of a stream of octets (bytes) between applications running on hosts communicating via an IP network. Major internet applications such as the World Wide Web, email, remote administration, and file transfer rely on TCP.

UDP uses a simple connectionless communication model with a minimum of protocol mechanism. UDP provides checksums for data integrity, and port numbers for addressing different functions at the source and destination of the datagram. It has no handshaking dialogues, and thus exposes the user's program to any unreliability of the underlying network; there is no guarantee of delivery, ordering, or duplicate protection. If error-correction facilities are needed at the network interface level, an application may use Transmission Control Protocol (TCP) or Stream Control Transmission Protocol (SCTP) which are designed for this purpose. UDP is suitable for purposes where error checking and correction are either not necessary or are performed in the application; UDP avoids the overhead of such processing in the protocol stack. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for packets delayed due to retransmission, which may not be an option in a real-time system.

6.3 RTP, RTCP, RTSP

The Real-time Transport Protocol (RTP) is a network protocol for delivering audio and video over IP networks. RTP is used in communication and entertainment systems that involve streaming media, such as telephony, video teleconference applications including WebRTC, television services and web-based push-to-talk features.

RTP typically runs over User Datagram Protocol (UDP). RTP is used in conjunction with the RTP Control Protocol (RTCP). While RTP carries the media streams (e.g. audio and video), RTCP is used to monitor transmission statistics and quality of service (QoS) and aids synchronization of multiple streams. RTP is one of the technical foundations of Voice over IP and in this context is often used in conjunction with a signalling protocol such as the Session Initiation Protocol which establishes connections across the network.

RTP was developed by the Audio-Video Transport Working Group of the Internet Engineering Task Force (IETF) and first published in 1996 as IETF RFC 1889 [i.42], superseded by IETF RFC 3550 [i.43] in 2003.

The Real Time Streaming Protocol (RTSP) is a network control protocol designed for use in entertainment and communications systems to control streaming media servers. The protocol is used for establishing and controlling media sessions between end points. Clients of media servers issue VHS-style commands, such as play, record and pause, to facilitate real-time control of the media streaming from the server to a client (Video On Demand) or from a client to the server (Voice Recording).

The transmission of streaming data itself is not a task of RTSP. Most RTSP servers use the Real-time Transport Protocol (RTP) in conjunction with Real-time Control Protocol (RTCP) for media stream delivery. However, some vendors implement proprietary transport protocols. RTSP was standardized by the Multiparty Multimedia Session Control Working Group of the Internet Engineering Task Force (IETF) and published as IETF RFC 2326 [i.44] in 1998. RTSP 2.0 published as IETF RFC 7826 [i.45] in 2016 as a replacement of RTSP 1.0. RTSP 2.0 is based on RTSP 1.0 but is not backwards compatible other than in the basic version negotiation mechanism.

6.4 WebRTC

W3C WebRTC [i.58] is a candidate recommendation of the World Wide Web Consortium. It defines means for real-time peer-to-peer data transfer between applications running in separate browsers. The transferred data can be arbitrary data as well as video and audio streams.

WebRTC may be of interest for AR applications running in a web browser. The reason is that the transferred stream data can be media streams captured with the HTML5 Media Capture and Streams recommendation. This means, WebRTC allows to capture the real world with one device and to display the captured data on another device. During the data transfer, adapted implementations of the recommendation may also adapt the data so that enriching the captured data may become possible.

7 Standards for Hardware API

7.1 Introduction

Fine control of AR devices is a requirement for many AR applications and services, due to the real time nature of the processing of data from sensors or of data to be rendered. Standards commonly used for hardware API are published by the Khronos Consortium and are introduced in the following clauses.

7.2 OpenVX™ specifications

OpenVX™ [i.64] is an open-royalty-free standard for cross platform acceleration of computer vision applications. OpenVX™ enables performance and power-optimized computer vision processing, especially important in embedded and real-time use cases such as face, body and gesture tracking, smart video surveillance, advanced driver assistance systems (ADAS), object and scene reconstruction, augmented reality, visual inspection, robotics and more.

OpenVX™ provides developers with a unique interface to design vision pipelines, whether they are embedded on desktop machines, on mobile terminals or distributed on servers. These pipelines are expressed thanks to an OpenVX™ graph connecting computer vision functions, called "Nodes", implementations of abstract representations called Kernel. These nodes can be coded in any language and optimized on any hardware as long as they are compliant with OpenVX™ interface. To improve performance, implementations of the graphs can fuse some nodes to eliminate memory transfers, can process at tile granularity instead of image granularity to parallelize heavy processing, can be split across the whole system (CPU/HPU/dedicated hardware) and can optimize memory by reusing pre-allocated memory for multiple intermediate data. As an example, these optimizations can improve by a factor of 8 the execution performance in terms of processing time of an image filter compared to an OpenCV implementation based on CUDA.

OpenVX™ provides developers with a set of data structures to transfer information between nodes (Dataflow):

- Array: any array of a primitive or array of structures.
- Convolution: a MxN matrix associated with a scaling factor.
- Delay: to manually delay list of objects.
- Distribution: to represent a frequency distribution such as a histogram.
- Image: a matrix of pixel with various formats.
- LUT: a lookup table representation.
- Matrix: a MxN matrix of scalar value.
- Pyramid: contains multiple levels of images.
- Remap: a map of source points to destination points used to transform an image.
- Scalar: a single primitive data type.
- Threshold: essentially used for configuration.
- ObjectArray: can be an array of any data-object.
- Tensor: a multidimensional data object used for Neural Networks.

Also, OpenVX™ provides developers with more than 60 vision operations interfaces (Gaussian image pyramid, Histogram, Optical flow, Harris corners, etc.) as well as conditional node execution and neural network acceleration (a list of these functions are available at https://www.khronos.org/files/openvx-12-reference-card.pdf).

OpenVX™ does not provide developers with all operations required to implement dedicated vision pipelines such as SLAM. Indeed, OpenVX™ API does not specify keypoints description extractors, keypoints matchers, perspective n points algorithms, bundle adjustment as well as pose graph structures. Nevertheless, third parties such as OpenCV provide C++ wrappers for OpenVX™ C API to interface vision operations. Moreover, OpenVX™ provides features to extend the OpenVX™ API by defining user kernels Also, discussions concerning the roadmap of OpenVX™ aim to provide developers with neural network import based on neural network files to embed CNN in Nodes as well as easy ways to let programmers extend the existing vision operations.

OpenVX™ is currently the only standard providing developers with hardware acceleration and power efficiency to implement real time vision based pipeline required by augmented reality applications (visual SLAM, object tracking, scene reconstruction, scene analysis, etc.). For this reason, OpenVX™ is a very good candidate for interfacing processing units with software dedicated to computer vision, but it will have to be extended to cover all the vision functions required by vision pipelines addressing augmented reality applications.

7.3 OpenXR™ specifications

OpenXR™ [i.65] defines two levels of API interfaces that a VR platform's runtime can use to access the OpenXR™ ecosystem. Applications and engines use standardized interfaces to interrogate and drive devices. Devices can self-integrate to a standardized driver interface. Standardized hardware/software interfaces reduce fragmentation while leaving implementation details open to encourage industry innovation. For areas that are still under active development, OpenXR™ also supports extensions to allow for the ecosystem to grow to fulfil the evolution happening in the industry.

The OpenXR™ working group aims to provide the industry with a cross-platform standard for the creation of VR/AR applications. This standard would abstract the VR/AR device capabilities (display, haptics, motion, buttons, poses, etc.) in order to let developers access them without worrying about which current hardware is used. In that way, an application developed with OpenXR™ would be compatible with several hardware platforms. OpenXR™ aims to integrate the critical performance concepts to enable developers to optimize for a single and predictable target instead of multiple proprietary platforms. OpenXR™ focuses on the software and hardware currently available and does not try to predict the future innovation of AR and VR technologies. However, its architecture is flexible enough to support such innovations in a close future.

Regarding the different elements of the current architecture, OpenXR™ has two main faces:

- The OpenXR™ "application interface" is an API that each VR or AR application or engine has to use in order to allow it to talk with the OpenXR™ runtime. This is the first level of abstraction (Vendor SDKs abstraction).
- The OpenXR™ "device plugin extension" lets multiple arbitrary devices talk with the OpenXR™ runtime. This is the second level of abstraction (device abstraction). This OpenXR™ device plugin extension is optional but this is the key for applications that want to be compatible with multiple devices. For instance an OpenXR™ runtime based on OpenVR could implement a device plugin extension to support different HMDs.

The OpenXR™ workflow can be described as follows:

- The application interface performs the rendering and transmits its image(s) to the OpenXR™ runtime.
- The OpenXR™ runtime processes these images, handles composition, distortions and collects haptics events and then sends them to the device(s). At the same time this runtime collects the tracking and input information from the device(s) and sends them to the application interface. A runtime is vendor specific but it does not mean that only hardware vendors can implement them.
- The application interface receives the device's inputs and adapts its state as a consequence.

Multiple kind of data are abstracted in OpenXR™. First, the viewport(s) of the currently used display system are abstracted in an object that can provide display size, view transform, projection specification and even gaze tracking data (for foveated rendering for instance). OpenXR™ can support multiple viewports including single viewport (example: Camera Passthrough AR), two viewports (Stereoscopic VR/AR HMD) and even more such as twelve viewports for a 6 faces CAVE system. Second, an application can access tracking data referred in OpenXR™ as "XrSpace" which contains a transformation matrix. For instance, it can refer to the head or a hand pose. Third, another layer of abstraction is built around actions ("XrActions") which include both inputs and haptics. This abstraction allows developers to define input based on resulting action ("Move", "Jump", "Teleport"). The bindings with concrete devices can be suggested by the application or directly handled by the runtime through user settings. A system based on semantic path is used to access the different inputs and outputs of OpenXR™. For instance, "spaces/hand/left" can be used to access the left hand space and "/user/hand/left/input/trigger/click" can be used to access the trigger button.

OpenXR™ abstracts the main relations between a VR/AR applications and concrete hardware, including access to inputs, haptics, to tracking information and the management of the complex VR/AR rendering pipeline (viewport and projection configuration, stereoscopy, compositor). This abstraction allows developers to target multiple hardware with the same API and therefore to reduce their development times and costs. Regarding specifically AR issues, the standard does not aim to standardize any AR tracking pipelines. However, it will allow to develop AR content without knowing which kind of AR display and tracking technologies are used. The support for future tracking display technology such as non-planar displays and the need for eye-display tracking calibration is not very clear for now. However, the standard seems flexible enough to support these future innovations as long as their designers propose their contributions to the standard.

7.4 OpenGL™ specifications

Khronos OpenGL [i.66] specification describes an abstract API for drawing 2D and 3D graphics. Although it is possible for the API to be implemented entirely in software, it is designed to be implemented mostly or entirely in hardware.

OpenGL is the premier environment for developing portable, interactive 2D and 3D graphics applications. Since its introduction in 1992, OpenGL has become widely used in the industry and supports 2D and 3D graphics application programming interface (API), bringing thousands of applications to a wide variety of computer platforms. OpenGL fosters innovation and speeds application development by incorporating a broad set of rendering, texture mapping, special effects, and other powerful visualization functions. Developers can leverage the power of OpenGL across all popular desktop and workstation platforms, ensuring wide application deployment.

Any visual computing application requiring maximum performance-from 3D animation to CAD to visual simulation-can exploit high-quality, high-performance OpenGL capabilities. These capabilities allow developers in diverse markets such as broadcasting, CAD/CAM/CAE, entertainment, medical imaging, and virtual reality to produce and display incredibly compelling 2D and 3D graphics.

7.5 WebGL™ specifications

Khronos WebGL™ [i.67] is a cross-platform, royalty-free web standard for a low-level 3D graphics API based on OpenGL™ ES, exposed to ECMAScript via the HTML5 Canvas element. Developers familiar with OpenGL™ ES 2.0 will recognize WebGL™ as a Shader-based API, with constructs that are semantically similar to those of the underlying OpenGL™ ES API. It stays very close to the OpenGL™ ES specification, with some concessions made for what developers expect out of memory-managed languages such as JavaScript. WebGL™ 1.0 exposes the OpenGL™ ES 2.0 feature set; WebGL™ 2.0 exposes the OpenGL ES 3.0 API.

7.6 Vulkan™ specifications

Khronos Vulkan [i.68], formerly named the "Next Generation OpenGL Initiative" (glNext), is a grounds-up redesign effort to unify OpenGL™ and OpenGL™ ES into one common API that will not be backwards compatible with existing OpenGL™ versions. The initial version of Vulkan API was released on February 16, 2016.

Vulkan is a low-overhead, cross-platform 3D graphics and computing API. Vulkan targets high-performance real-time 3D graphics applications such as video games and interactive media across all platforms. Compared to OpenGL™ and Direct3D 11, and like Direct3D 12 and Metal, Vulkan is intended to offer higher performance and more balanced CPU/GPU usage. Other major differences from Direct3D 11 (and prior) and OpenGL are Vulkan being a considerably lower level API and offering parallel tasking. Vulkan also has the ability to render 2D graphics applications. In addition to its lower CPU usage, Vulkan is also able to better distribute work among multiple CPU cores. Vulkan is said to induce anywhere from a marginal to polynomial speedup in run time relative to other APIs if implemented properly on the same hardware.

8 User interaction standards

8.1 Gestures

8.1.1 Introduction

Current AR devices often provide computer vision based technologies to enable contactless gesture control of augmented objects, GUIs or settings of the AR device itself.

8.1.2 DIN SPEC 91333

In 2016 German DIN published the DIN SPEC 91333 [i.69] - Contactless gesture control for human-system interaction to provide instructions and recommendations for designing systems controlled by the means of intended or unintended body gestures performed in a three dimensional space.

Components of this DIN SPEC are central terms, a representation of the process of contactless gesture control and the description, labelling and presentation (illustration) of human gestures. Furthermore, basic rules for designing usable gestures are defined and gesture examples are presented. DIN SPEC is neither a comprehensive catalogue of gestures nor a comprehensive list of applications, since they have to be created for specific industries or applications. This DIN SPEC is aimed at developers, product manufacturers, buyers, testers and end users of gesture-controlled systems. This DIN SPEC should be applied together with DIN EN ISO 9241-960 [i.70] and ISO/IEC 30113-1 [i.22] and deals exclusively with the contactless gesture control specific aspects of the user interface in human-system interaction.

Since this form of interaction depends on the task, the user, the context and the technology, this DIN SPEC should not be applied without knowledge of these variables or as a fixed specification. Rather, it is assumed that developers and designers have the relevant information regarding the task, user, environment and technology for the design of the human-system interface through contactless gesture control.

8.1.3 ISO MPEG-U

ISO/IEC 23007-2 [i.23] called MPEG-U Advanced User Interaction (AUI) interface aims to support various advanced user interaction devices. The AUI interface is part of the bridge between scene descriptions and system resources. A scene description is a self-contained living entity composed of video, audio, 2D graphics objects, and animations. Through the AUI interfaces or other existing interfaces such as DOM events, a scene description accesses system resources of interest to interact with users. In general, a scene composition is conducted by a third party and remotely deployed. Advanced user interaction devices such as motion sensors and multi touch interfaces generate the physical sensed information from user's environment. By a recognition process, a set of physical information can be converted to a pattern with semantics which is more useful to a scene description. For instance, some feature points drawn by user's finger can be understood as a circle which is specified with the centre position of a circle and a radius value. Therefore, this part provides a set of data formats which defines geometric patterns, symbolic patterns, touch patterns, posture patterns and their composite patterns.

The geometric patterns are a set of geometric shapes which are recognized with sensed geometric information as 2D or 3D Cartesian positions. Current version of standard defines the following geometric patterns: Point, Line, Rectangle, Arc and Circle. Instead of speaking or writing words, simple well-known gestures help to communicate with others. For instance, V sign and Rock sign, which have well-known common semantics, are already used in various circumstances. Therefore, the standard provides a container format for symbolic patterns and a classification scheme to enlist well-known symbolic shapes.

Many applications adopt well-known touch patterns for users who use touch-based interaction devices. In this standard, a container format for well-known touch patterns and a classification scheme to enlist basic touch patterns are provided. Also, these touch patterns can be captured via not only touch-based interaction devices but also other intelligent devices.

This standard also describes the hand posture patterns to support the intuitive hand-based interaction for scene description. For example, if a user wants to control an object in a scene description, the user in the real life does a hand posture such as grabbing, fist, and open palm which would be a good candidate gesture to support such an interaction modality.

9 Domain-specific standards

9.1 Building/construction

The Industry Foundation Classes (IFC) (ISO/IEC 16739 [i.38]) is a platform neutral, open file format specification that intends to simplify the data exchange in the building and construction industry. IFC cannot be transferred directly into AR/VR technologies due to its geometric representation. For the transfer of an IFC-File to AR/VR development environments, the given geometries from IFC is converted into triangulated meshes. For this purpose, parsers can be developed independently with common IFC toolkits. Afterwards the extracted mesh information can be converted into suitable formats such as COLLADA DAE or FBX (these are the only formats that ensure the hierarchical structure of 3D-objects). An existing solution for converting IFC to COLLADA format is *IfcOpenShell*. Due to the considerable changes as a result of extensions and further developments of the IFC versions, independently created parsers have to be adapted regularly.

An alternative for the transmission of building data in AR/VR devices is the already implemented solution of the Autodesk Forge AR/VR Toolkit. This solution provides that formats which are frequently used in the building industry can be uploaded into a cloud platform, translated into the internal format and displayed as a hierarchical scene, e.g. in the AR development environment of a programming tool. Unlike IFC, the formats used by Autodesk Forge are not manufacturer-neutral.

9.2 Manufacturing and Installations

9.2.1 Introduction

Installation tasks share a lot of features with maintenance, and standards often apply to both kinds of work. But the range of these standards is wide and the scope of each is broad. They are also often differentiated by industry branches, and there are indeed a lot of particularities for such different industries as electronics, electricity, construction, chemistry, oil, agriculture, food or manufacturing.

In order to narrow our field of investigation this clause focuses on a process that is common to most industries; the workflow of work orders as shown in Figure 6. The work order workflow can take many forms but the simple following breakdown can be used.

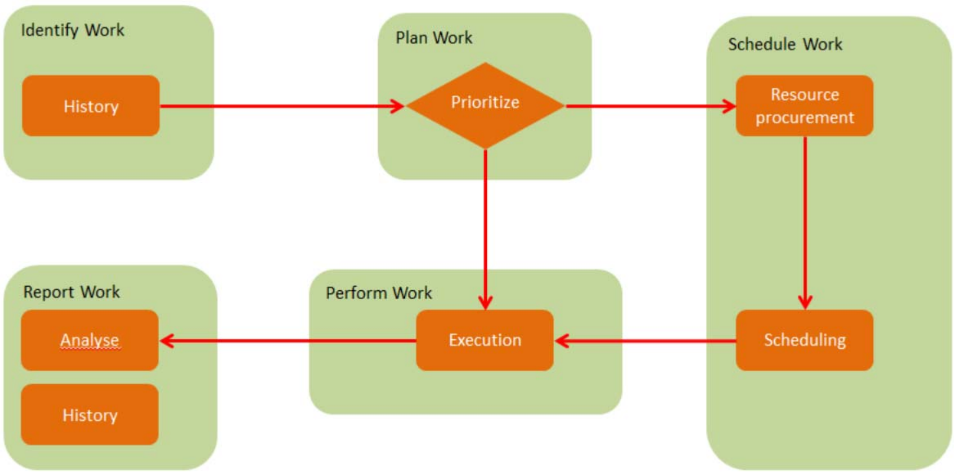

Figure 6: Workflow of work orders

This process consists of a long succession of related tasks. AR technologies are primarily relevant for the execution of the work order, but it can also play a role during the work preparation and the work completion.

Relevant information should be presented to the technicians in a Head Up display in order to let them work hand free; sequences of actions for procedures, check lists, reports and many kinds of textual data.

9.2.2 Process sequence

Work to be done reaches technicians through work orders. Before the actual installation or maintenance operation phase, the technician needs to get the work orders, understand them and prepare the technical equipment to perform the planned tasks as described in table 5.

Table 5: Sequencing a work process

Understanding and preparing	Depending on the task to achieve, the technician can also try and do the operations in AR before performing them on the real critical equipment. This can be done before a very sensitive operation but also directly on the equipment to install or maintain.
Materials and resource procurement	The process of guiding the technician through a storeroom or the picking of equipment and tools could also be assisted through AR.
Navigating to the place of work	Augmented reality features allow guiding people through a known environment and can help improve access to equipment especially on large installations.
Augmenting equipment and machines	Perform AR to overlay information over an equipment, to highlight parts of it to show points of interest for quality assurance, or to display the equipment status.
Showing the sequence of tasks to execute	Guiding the technician to execute each step of a complex procedure in the right order by superimposing them over the real equipment.
Annotating reality in AR	Allowing the operator to annotate and/or capture a situation to share or record his observations.

The execution of the work order is connected to some relevant standards allowing the connection with Computerized Maintenance Management Systems (CMMS) or Enterprise Resource Planning (ERP) solutions. In the following sections, some of these standards are introduced, however, they cover a very wide spectrum of aspects, not necessarily directly related to augmented reality technologies.

9.2.3 Enterprise asset management

Enterprise Asset Management (EAM) involves the management of the maintenance of physical assets of an organization throughout each asset's lifecycle. EAM is used to plan, optimize, execute, and track the maintenance activities with the associated priorities, skills, materials, tools, and information. This covers the design, construction, commissioning, operations, maintenance and decommissioning or replacement of plant, equipment and facilities.

It is described by two main references, one from ASTM and one from ISO, both introduced in table 6.

Table 6: References for Enterprise asset management

ASTM International https://www.astm.org	ASTM International, formerly known as American Society for Testing and Materials, is an international standards organization that develops and publishes voluntary consensus technical standards for a wide range of materials, products but also for systems, and services. They edit many standards about best practices and processes about assets management. The documents are not free to read, but here are three standard examples that could be of interest.	ASTM E2132 - 17 [i.71]: Standard Practice for Inventory Verification: Electronic and Physical Inventory of Assets ASTM E3035 - 15 [i.72]: Standard Classification for Facility Asset Component Tracking System ASTM E2499 - 18 [i.73]: Standard Practice for Classification of Asset Location Information
ISO 55000 [i.24] series	ISO 55000 [i.24] is an international standard covering management of assets of any kind. Before it, a Publicly Available Specification (PAS 55) was published by the British Standards Institution in 2004 for physical assets. The ISO 55000 [i.24] series of Asset Management standards was launched in January 2014. This standard is more about requirements checklist of good practices in physical asset management than technical requirements.	ISO 55000 [i.24] provides critical overview, concepts and terminology ISO 55001 [i.25] specifies the requirements for an effective Asset Management System ISO 55002 [i.26] offers interpretation and guidance for such a system to be implemented.

9.2.4 Computerized Maintenance Management System

A computerized maintenance management system (CMMS) is a software package designed to maintain a computer database for an organization's maintenance operations and human resources functions. This data is intended to help the effectiveness of maintenance workers, the quality of management decisions and the verification of regulatory compliance. CMMS is also closely linked with Enterprise Asset Management and Enterprise Resource Planning. Several standards covering this domain are published, as illustrated in table 7.

Table 7: References for Computerized Maintenance Management System

ISO 15926-1 [i.27]	ISO 15926-1 [i.27] is a standard for data integration, sharing, exchange, and hand-over between computer systems. It contains several parts that seem of interest, but the standard is not freely accessible.	• Part 1: Introduction. Information concerning engineering, construction and operation of production facilities is created, used and modified by many different organizations throughout a facility's lifetime. The purpose of ISO 15926-1 [i.27] is to facilitate integration of data to support the lifecycle activities and processes of production facilities. • Part 2: Data Model. A generic 4D model that can support all disciplines, supply chain company types and life cycle stages, regarding information about functional requirements, physical solutions, types of objects and individual objects as well as activities. • Part 3: Reference data for geometry and topology. Specifies geometric and topological concepts, enabling the recording of geometric and topological data using ISO 15926-2 [i.28]. Also specifies concepts related to mesh topology and functions defined with respect to meshes, enabling the recording of mesh topology data and the representation of property distributions. • Part 9 (in development)- Implementation standards, with the focus on Façades, standard web servers, web services, and security. • Part 12 (in development)- Life cycle integration ontology in Web Ontology Language (OWL2).

9.2.5 Other standards

The three organizations listed in table 8 work together to the construction of coherent standards related to operations & maintenance.

Table 8: Standardization bodies in the area of operations and maintenance

MIMOSA - Operations and Maintenance Information Open System Alliance http://www.mimosa.org/	MIMOSA is a not-for-profit trade association dedicated to developing and encouraging the adoption of open information standards for Operations and Maintenance in manufacturing, fleet, and facility environments. MIMOSA's open standards enable collaborative asset lifecycle management in both commercial and military applications.	• MIMOSA CCOM 4.0.0 [i.74] (Common Collaborative Object Model) serves as an information model for the exchange of asset information. Its core mission is to facilitate standards-based interoperability between systems: providing an XML model to allow systems to electronically exchange data. • MIMOSA OSA-EAI [i.75] (Open System Architecture for Enterprise Application Integration) provides an information exchange standard to allow sharing asset registry, condition, maintenance and reliability information between enterprise systems and a relational database model to allow storage of the same asset information.
OpenO&M - Open operation & maintenance initiative http://www.openoandm.org/about/	OpenO&M is a nonprofit organization dedicated to the development and use of a harmonised set of standards for the exchange of Operations & Maintenance data.	

OPC foundation - Open Platform Communications https://opcfoundation.org/	The OPC Foundation (OPC formally known as Object Linking and Embedding for Process Control) is an industry consortium that creates and maintains standards for open connectivity of industrial automation devices and systems, such as industrial control systems and process control generally. The OPC standards specify the communication of industrial process data, alarms and events, historical data and batch process data between sensors, instruments, controllers, software systems, and notification devices.	

10 Standards for assessing AR quality

10.1 User experience

Two important factors for AR systems are their user experience and their usability. This encompasses the technical setup as well as the applications. User experience focuses on the user's emotions and attitudes about a product throughout its lifecycle. This covers the intention to buy or use a product, the product usage itself, as well as any subsequent product related aspects, e.g. handling of defects or the disposal of a product. Usability focuses only on the actual usage of a product. Important in this context are the tasks that users want to perform with a product. A good usability is provided if users can accomplish their tasks with the product with effectiveness, efficiency, and satisfaction. Considering an AR system as a product, both terms are applicable for AR, and usability may be of greater importance for industrial AR applications.

The most referenced definitions of user experience and usability are provided by the ISO standard series 9241 (ISO 9241 [i.29]) called "Ergonomics of Human System Interaction". This standard consists of multiple parts each focusing on different aspects. The definition of user experience is given in part 210. This part focuses on "Human-centred Design Processes for Interactive Systems". Usability is defined in part 11 called "Guidance on Usability". For AR systems, other parts of ISO standard 9241 may also be applicable. For example, part 125 provides "Guidance on Visual Presentation of Information" and part 161 contains "Guidance on Visual User Interface Elements".

A similar definition of usability to ISO 9241 [i.29] is provided by ISO 9126 [i.30]. This standard focuses on quality aspects of software in general and categorizes these aspects into multiple groups. The name of one of these groups is Usability. The corresponding definition in the ISO 9126 [i.30] standard has been adapted in its latest version for a better matching to the definition in ISO 9241 [i.29].

To evaluate the user experience and the usability of systems, diverse methods exist. The technical report ISO/TR 16982:2002 provides an overview of available methods as well as a proposal of when and how to apply these methods during the software development cycle. The methods listed in that report focus strongly on usability. As usability also contributes to user experience, these methods are helpful for both areas.

For reporting on usability evaluations, there is the Common Industry Format (CIF). It specifies what needs to be considered for usability evaluation preparations and result documentation. The CIF is specified in multiple ISO technical reports. ISO/IEC 25060 [i.31] introduces the CIF and provides a corresponding overview. ISO/IEC 25062 [i.32] defines in more concrete terms what belongs to a reporting of usability findings developed by quantitative usability evaluation methods. The definition of usability shows that a good usability is only given for a certain context of use. This context includes the groups of users, their goals and tasks, as well as the physical environment of product usage. The technical report ISO/IEC 25063 [i.33] provides guidance on how to describe and specify details on the context of use of a software. In addition, ISO/IEC 25064 [i.34] specifies how to document further user needs.

For users, it is always good to recognize known things when working with new systems. This helps users to understand more easily how something new may work and increases the learnability of systems. Hence, also AR systems should reuse known concepts. Examples for this are well-known icon sets or approaches of how to handle multimedia in the application. An example for a standard on how to handle multimedia is ISO 14915 [i.35].

A term related to usability and user experience is accessibility. Accessibility aims at allowing any user with any kind of impairment to use a system. A well-known example is the support for screen-readers to allow people with visual impairments to work with a website. But impairments are much more manifold and so is the consideration of accessibility. One important standard on accessibility is the Web Content Accessibility Guideline provided by the W3C consortium. This guideline is now an ISO standard under the number ISO/IEC 40500:2012 [i.36].

Annex A:
Authors & contributors

The following people have contributed to the present document:

Rapporteur:
Dr. Marius Preda, Institut Mines Telecom

Other contributors:

Ms. Muriel Deschanel, b<>com

Dr. Patrick Harms, Institute of Computer Science, University of Goettingen

Dr. Jens Johann, Deutsche Telekom AG

Mr. Traian Lavric, Institut Mines-Telecom

Ms. Nicole Le Minous, b<>com

Ms. Christine Perey, Perey Research & Consulting

Dr. Jerome Royan, b<>com

Dr. Ralf Schäfer, Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute

Mr. Eric Villain, Orange

Annex B: Bibliography

- EPRI 2017: "Program on Technology Innovation: Enterprise Augmented Reality Vision, Interoperability Requirements, and Standards Landscape", Technical Report, 3002010514.
- Milgram, P., Takemura, H., and Utsumi, A. and Kishino, F. 1994: "Augmented reality: A class of displays on the reality-virtuality continuum", Proc. of Tele-manipulator and Telepresence Technologies, pp. 2351-34.
- Azuma, R. 1997: "A survey of augmented Reality. Presence: Teleoperators and Virtual Environments", 6 (4), pp. 355 - 385.
- Billinghurst, M. Kato, H. and Poupyrev, I. 2001: "The MagicBook: A transitional AR interface", Computers and Graphics, 25 (5), pp. 745-753.
- Cheok, A., Goh, K., Liu, W., Farbiz, F., Fong, S., Teo, S., Li, Y. and Yang, X. 2004: "Human pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing", Personal and Ubiquitous Computing, 8, pp. 71-81.
- Thomas, C., Close, B., Donoghue, J., Squires, J., De Bondi, P., Morris, M. and Piekarski, W. 2000: "ARQuake: An outdoor/indoor augmented reality first person application", Proc. of Intl. Symposium on Wearable Computing, pp. 139-146.
- Linderman, R., Noma, H. and de Barros, P. 2008: "An empirical study of hear-through augmented reality: using bone conduction to deliver spatialized audio", Proc. of IEEE Virtual Reality, pp. 35-42.
- Lee, G., Dunser, A., Kim S. and Billinghurst, M. 2012: "CityViewAR: A mobile outdoor AR application for city visualization", Proc. of Intl. Symposium on Mixed and Augmented Reality, pp. 57-64.
- Bandyopadhyay, D., Raskar, R. and Fuchs, J. 2001: "Dynamic shader lamps: Painting on movable objects", Proc. of Intl. Symposium on Augmented Reality, pp. 207-216.
- Klein, G. and Murray, D. 2007: "Parallel tracking and mapping for small AR workspaces", Proc. of Intl. Symposium on Mixed and Augmented Reality, pp. 1-10.
- Davison, A., Reid, I., Molton, N. and Stasse, O. 2007: "MonoSLAM: Realtime, single camera SLAM". IEEE Trans. On Pattern Analysis and Machine Intelligence, 29(6), pp. 1052-1067.
- Newcombe, R., Izadi, S., Hillinges, O., Molyneaux, D. and Kim, D., Davison, A., Kohli, P., Shotton, J., Hodges, S. and Fitzgibbon, A. 2011: "KinectFusion: Real time dense surface mapping and tracking", Proc. of Intl. Symposium on Mixed and Augmented Reality, pp. 127-136.
- Lavric, T., Scurtu, V. and Preda, M. 2014: "Create and play augmented experiences", Technical report from the 104th MPEG meeting, Inchon.
- Hill, A., Macintyre, B., Gandy, M., Davidson, B. and Rouzati, H. 2010: "Khamra: An open KML/HTML architecture for mobile augmented reality applications", Proc. of Intl. Symposium on Mixed and Augmented Reality, pp. 233-234.
- Kato, H. and Billinghurst, M. 1999: "Marker tracking and hmd calibration for a video-based augmented reality conferencing system", Proc. of Intl. Workshop on Augmented Reality, pp. 85-94.

History

Document history		
V1.1.1	April 2019	Publication