ETSI GS ARF 003 V1.1.1 (2020-03)

GROUP SPECIFICATION

Augmented Reality Framework (ARF); AR framework architecture

Disclaimer

Reference

DGS/ARF-003

Keywords

API, architecture, augmented reality, context capturing and analysis, framework, model, real time

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 Specification (GS) 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 document specifies a functional reference architecture for AR solutions.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**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 specifies a functional reference architecture for AR components, systems and services. The structure of this architecture and the functionalities of its components have been derived from a collection of use cases ETSI GR ARF 002 [i.3] and an overview of the current landscape of AR standards ETSI GR ARF 001 [i.4].

The present document introduces the characteristics of an AR system and describes the functional building blocks of the AR reference architecture and their mutual relationships. The generic nature of the architecture is validated by mapping the workflow of several use cases to the components of this framework architecture.

2 References

2.1 Normative 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.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

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 necessary for the application of the present document.

Not applicable.

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] Ronald T. Azuma, "A survey of augmented reality", in Presence: Teleoperators & Virtual Environments, 1997.

[i.2] Paul Milgram, H. Takemura, A. Utsumi, F. Kishino, "Augmented Reality: A class of displays on the reality-virtuality continuum", in Proceedings of Telemanipulator and Telepresence Technologies, 1994.

[i.3] ETSI GR ARF 002 (V1.1.1): "Augmented Reality Framework (ARF) Industrial use cases for AR applications and services".

[i.4] ETSI GR ARF 001 (V1.1.1): "Augmented Reality Framework (ARF); AR standards landscape".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

Augmented Reality (AR): ability to mix in real-time spatially-registered digital content with the real world

NOTE: This definition is based on the work of Azuma [i.1] and Milgram [i.2].

AR anchor: coordinate system related to an element of the real world on which virtual content stays spatially-registered

NOTE: AR anchors help users to maintain the perception that static virtual content appears to stay at the same position and orientation in the real world.

AR application: software designed by using several **AR components** to perform a group of coordinated functions, tasks, or activities for the benefit of the user who is experiencing augmented reality

AR component: hardware or software that provides application-oriented computing functions and supports interoperability when connected with other components of the **AR system**

AR device: hardware that provides one or more functions offering an augmented reality experience to one or several users

AR experience: the real time perception of the mixture of the real world and spatially-registered digital content by user senses

AR system: combination of hardware and software that delivers an AR experience

AR scene: information describing the interactive content contributing to an augmented reality experience

Building Information Modeling (BIM): process supported by various tools and technologies involving the generation and management of digital representations of physical and functional characteristics of places

descriptor extraction: task consisting in extracting differentiating characteristics of a detected feature

feature detection: task consisting in detecting specific information from a given signal

function: collection of functionalities

Product Lifecycle Management (PLM): process of managing the entire lifecycle of a product from inception through engineering, design, and manufacture to service and disposal of manufactured products

pose: position and orientation of an object, defined in a given coordinate system

EXAMPLE: The camera pose defined in a world coordinate system.

pose estimation: task of determining the pose of an object

object recognition: task consisting in finding and identifying objects

EXAMPLE: Recognition may be performed on an image, a video sequence, or an audio stream.

object tracking: task consisting in locating an object over time

EXAMPLE 1: A 2D tracking consists in locating an object in a sequence of images.

EXAMPLE 2: A 3D tracking consists in locating an object in a 3D space from a sequence of images or an audio signal.

point cloud: set of data points in space defined in a common coordinate system

EXAMPLE: A 3D point cloud is a set of data points in a 3D space.

random forest: learning method based on a multitude of decision trees used for classification or regression tasks

reference point: point located at the interface of two non-overlapping functions and representing interrelated interactions between those functions

visual bag of words: simplified representation using image features as words for image retrieval task

visual descriptor: characteristics of a visual feature

NOTE: A descriptor is based on elementary characteristics such as the shape, the colour, the texture or the motion of the feature itself and its neighbourhood in the image.

EXAMPLE: SIFT, SURF, BRIEF, ORB, BRISK, FAST, etc.

visual feature: information representing an element of an image

NOTE: The feature are generally primitive geometric elements (points, edges, lines, polygons, colors, textures, or any shapes) used to characterize an image.

EXAMPLE: Keypoints, edges, blobs.

3.2 Symbols

Void.

3.3 Abbreviations

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

AR	Augmented Reality
AV	AudioVisual
BIM	Building Information Modeling
BRIEF	Binary Robust Independent Elementary Features
CAD	Computer-Aided Design
DoF	Degree of Freedom
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphic processing Unit
GUI	Graphical User Interface
IMU	Inertial Measurement Unit
IP	Internet Protocol
Li-Fi™	Light Fidelity
ORB	Oriented FAST and Rotated Brief
PLM	Product Lifecycle Management
RGB	Red, Green, Blue
RGB-D	Red, Green, Blue and Depth
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
TPU	Tensor Processing Unit
UWB	Ultra-WideBand
VPU	Vision Processing Unit
Wi-Fi™	Wireless Fidelity

4 Architecture Overview

4.1 Global Architecture

An AR system is based on a set of hardware and software components as well as data describing the real world and virtual content. Figure 1 presents a global overview of an AR system architecture. The architecture diagram is structured in three layers, in the upper part the hardware, in the middle the software, and in the lower part the data:

- Hardware layer:
  - **Tracking Sensors:** These sensors aim to localize (position and orientation) the AR system in real-time in order to register virtual contents with the real environment. Most of AR systems such as smartphones, tablets or see-through glasses embed at least one or several vision sensors (generally monochrome or RGB cameras) as well as an inertial measurement unit and a GPS™. However, specific and/or recent systems use complementary sensors such as dedicated vision sensors (e.g. depth sensors and event cameras), or exteroceptive sensors (e.g. Infrared/laser tracking, Li-Fi™ and Wi-Fi™).
  - **Processing Units:** Computer vision, machine learning-based inference as well as 3D rendering are processing operations requiring significant computing resources optimized thanks to dedicated processor architectures (e.g. GPU, VPU and TPU). These processing units can be embedded in the device, can be remote and/or distributed.
  - **Rendering Interfaces:** Virtual content require interfaces to be rendered to the user so that he or she can perceive them as part of the real world. As each rendering device has its own characteristics, the signals generated by the rendering software generally need to be transformed in order to adapt them to each specific rendering hardware.
- Software layer:
  - **Vision Engine:** This software aims to mix the virtual content with the real world. It consists of localizing (position and orientation) the AR device relative to the real world reference, localizing specific real objects relatively to the AR device, reconstructing a 3D representation of the real world or analysing the real world (e.g. objects detection, segmentation, classification and tracking). This software component essentially uses vision sensors signals as input, but not only (e.g. fusion of visual information with inertial measurements or initialization with a GPS), it benefits from the hardware optimization offered by the various dedicated processors embedded in the device or remote, and will deliver to the rendering engine all information required to adapt the rendering for a consistent combination of virtual content with the real world.
  - **3D Rendering Engine:** This software maintains an up-to-date internal 3D representation of the virtual scene augmenting the real world. This internal representation is updated in real-time according to various inputs such as user's interactions, virtual objects behaviour, the last user viewpoint estimated by the Vision Engine, an update of the World Knowledge to manage for example occlusions between real and virtual elements, etc. This internal representation of the virtual content is accessible by the renderer (e.g. video, audio or haptic) which produces thanks to dedicated hardware (e.g. Graphic Processing unit) data (e.g. 2D images, sounds or forces) ready to be played by the Rendering Interfaces (e.g. screens, headphones or a force-feedback arm).
- Data layer:
  - **World Knowledge:** This World Knowledge represents the information either generated by the Vision Engine or imported from external tools to provide information about the real world or a part of this world (CAD model, markers, etc.). This World Knowledge corresponds to the digital representation of the real space used for different usages such as localization, world analysis, 3D reconstruction, etc.
  - **Interactive Contents:** These Interactive Contents represent the virtual content mixed to the perception of the real world. These contents can be interactive or dynamic, meaning that they include both 3D contents, their animations, their behaviour regarding input events such as user's interactions. These Interactive Contents could be extracted from external authoring tools requiring to adapt original content to AR application (e.g. 3D model simplification, fusion, and instruction guidelines conversion).

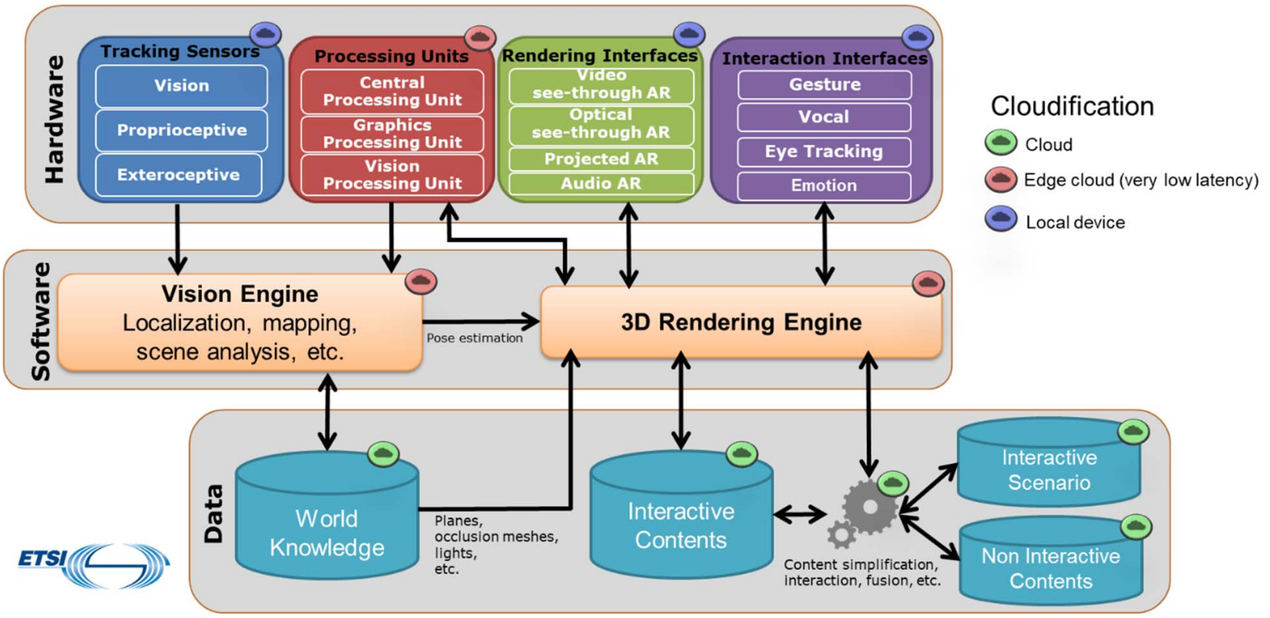

Figure 1: Global overview of the architecture of an AR system

4.2 Functional Architecture

Figure 2 shows the functional architecture specified by the present document addressing both fully embedded AR systems and implementations spread over IP networks in a scalable manner. Logical functions are shown as named boxes that may be nested in cases where a high-level function is composed of several subfunctions. The logical functions are connected by reference points. A reference point in a functional architecture is located at the conjunction of two non-overlapping functions and represents the interrelated interactions between those functions. A reference point allows a framework to aggregate those abilities that one function provides towards another function. In a practical deployment each of these reference points can be realized by a physical interface that conveys information between the connected subfunctions in a unidirectional or bidirectional way using a specified protocol. Depending on the deployment scenario and the applications that needs to be supported, multiple logical subfunctions can also be combined in one deployable unit. All of these subfunctions can either be deployed on the device that also presents the AR implementation or they can be provided via cloud technology.

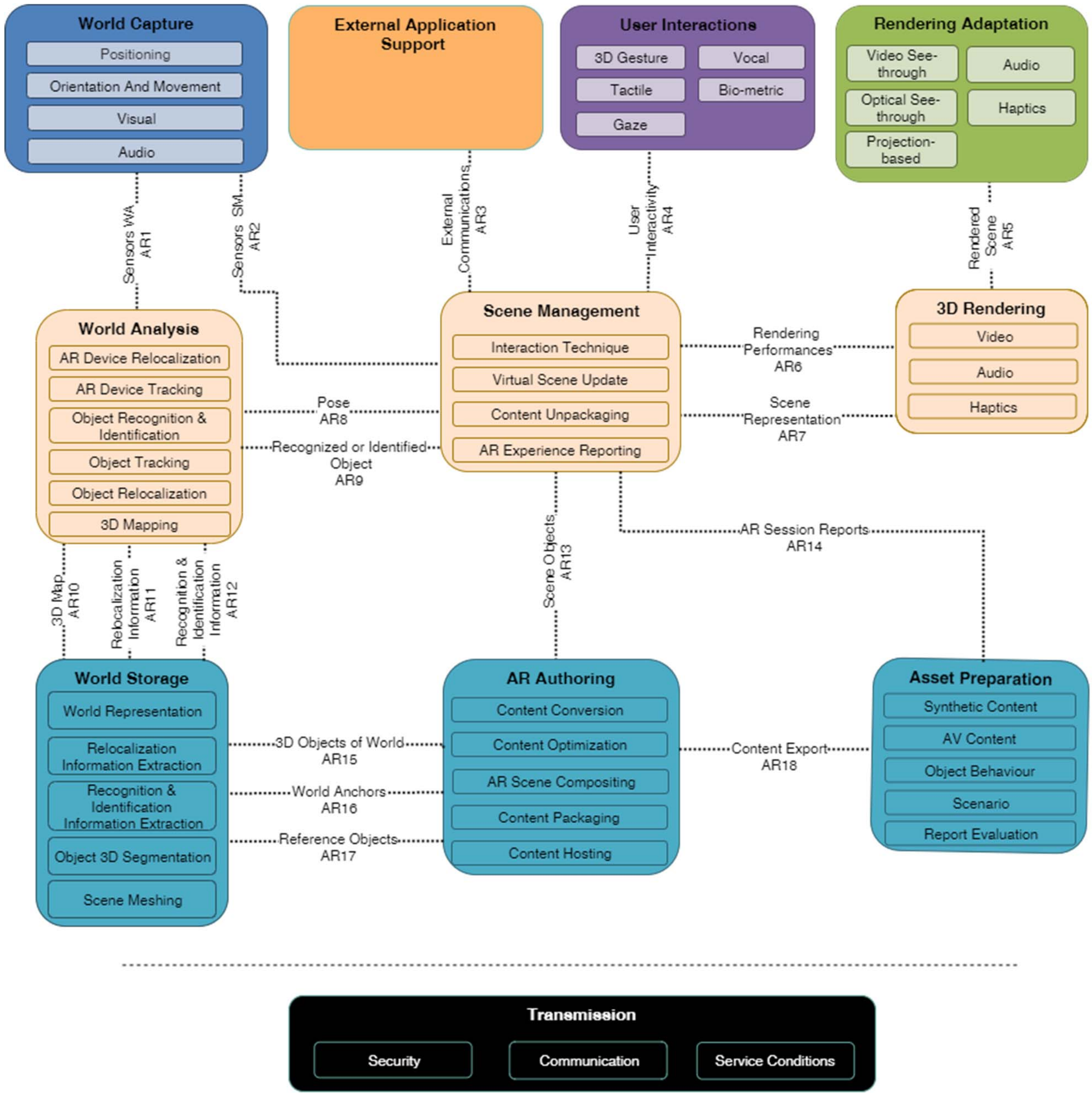

Figure 2: Diagram of the functional reference architecture

5 Functions and subfunctions of the functional architecture

5.1 World Capture

5.1.1 Introduction

This function deliver information relevant for the localization of the AR device or real objects, or for analysing the environment of the application. AR systems can embed various sensors aiming at better understanding the real environment as well as the pose (position and orientation) of the AR system or of real objects in this environment required to provide an accurate registration of virtual objects on the real world. The following subfunctions can address different kinds of sensors.

5.1.2 Positioning

This subfunction shall deliver the location of an AR device and may also deliver its orientation relatively to a coordinate reference system. The coordinate reference system shall be defined in relation to a reference in the real world and there shall be a mechanism to temporally synchronize this information with other **World Capture** subfunctions.

NOTE 1: The coordinate reference system can be related to the earth (global geo-positioning system), a factory, a room, an object, the positioning system itself, etc.

NOTE 2: GNSS are the most commonly used means to provide a position, but other solution such as beacons, Li-Fi™, UWB or other radio technologies can also provide a position or even an orientation of an AR system device.

5.1.3 Orientation and Movement

This subfunction shall deliver information about the movement or the orientation of the device that is providing the AR experience. This information shall be defined in a given coordinate system and there shall be a mechanism to temporally synchronize this information with other **World Capture** subfunctions.

NOTE 1: This subfunction can make use of the information provided by the subfunction **Positioning.**

NOTE 2: Inertial measuring can provide information about the movement and orientation of the device with a high frequency (~1 000 captures per seconds) which is useful to interpolate intermediate device poses between two vision-based pose estimations or when vision-based pose estimation fails.

5.1.4 Visual

This subfunction applies to AR systems making use of cameras (e.g. RGB, depth or event-based) which support the device providing AR experience. The cameras can be built into the device (interoceptive capture) or positioned outside the device (exteroceptive capture).

This subfunction shall deliver streaming video/event or still pictures to be used by the **World Analysis** function.

This subfunction should also deliver information about camera parameters relevant for the application (e.g. focal lengths, pixel size, principal point, image size, global or rolling shutter, distortion parameters, timing information and camera range) and in the case where the application uses several vision sensors, their relative positions and orientations.

Different kind of visual sensors can be addressed by this subfunction:

- RGB cameras make use of a photo-sensitive sensors by which coloured images are acquired. Colours are represented by an additive combination of the three primary colours red, green and blue.

- Depth cameras produce two-dimensional images that contain information about the distance to points of a scene from a given specific point. Several technologies can be used to achieve such information (e.g. stereoscopic triangulation, light patterns). In many cases, a depth camera can also provide a cloud of points defined in the coordinate system of the sensor.

- RGB-D cameras produce both two-dimensional images captured by a RGB camera and two-dimensional images captured by a depth sensor. For this reason, RGB-D cameras provide both camera and depth sensor interfaces. But, since an RGB-D camera is usually composed of two separate sensors, an RGB camera and a depth sensor (time of flight, structured-lights-based, stereoscopic, etc.), the raw images from the two sensors are not aligned. For this reason, RGB-D sensors offer complementary interfaces to match RGB and depth images.

- Event-based cameras measure the changes in brightness and colour in a scene over a given time period and allows the detection of movement of objects within the scene.

- Infrared cameras measure the infrared radiation of an object and maps the measured wavelength range into a picture using pseudo-colours.

- Laser trackers are used to measure the distance between such devices and objects that reflect laser pulses sent out by the tracker. Often, these measurements are mapped into a local coordinate system.

5.1.5 Audio

This subfunction applies to AR systems making use of microphones which support the device providing AR experience. The microphone can be built into the device (interoceptive capture) or positioned outside the device (exteroceptive capture).

This subfunction shall deliver audio streams which can be used by the World Analysis function for the scene compositing. These audio streams can be temporally and spatially aligned with video streams to capture not only the visual appearance but also the audio signals of a scene.

5.2 World Analysis

5.2.1 Introduction

This function defines the analysis processing of the real world. It can estimate the pose or movement of real objects or of the device providing the AR experience. It can also recognize or identify elements present in the real world, or it can reconstruct a 3D map of the real world. It can use video captures coupled with inertial measuring or positioning information to provide the AR service with localization, 3D reconstruction of the world, 3D object tracking or any other information extracted from sensors. Additional support can also be given by audio information.

As is the case in many AR implementations, first the registration of virtual content on the real world (the 3D alignment of real and virtual space) should be done. This can be achieved by positioning the virtual content relatively to virtual coordinate systems (anchors) attached to real elements (e.g. the world, objects or the AR device) based on the AR Scene Compositing subfunction of the AR Authoring function. Afterwards, the registration between virtual and real content should be obtained in real-time. This can be achieved by aligning the virtual viewpoint with the real one (the pose of the AR device) or the virtual coordinate system attached to objects (anchors) with their pose in the real world.

5.2.2 AR Device Relocalization

This subfunction shall deliver the pose (position and orientation) of the device providing the AR experience in a world coordinate system at present time without any prior knowledge of the pose of the device. It should use one or more subfunctions of the **World Capture** function. For example, it can use images, positions or orientations provided by respective subfunctions. Additional support can also be given by audio information. To provide this pose, the function may use a prior knowledge of the real environment provided by the World Storage function.

5.2.3 AR Device Tracking

This subfunction shall deliver the pose (position and orientation) of the device providing the AR experience in a world coordinate system at present time based on a prior knowledge built with previous frames (e.g. the pose of the AR device corresponding to the previous frame, a 3D structure reconstructed during the previous frames). It can analyse the motion between the current data captured by the AR device and the data previously captured (e.g. based on optical flow or feature matching). Additional support can also be given by audio information.

5.2.4 Object Recognition & Identification

This subfunction shall recognize or identify objects from the real world. It can use images, video streams, audio streams (audio-based object recognition or identification), an UWB-based identification system, or other data provided by the **World Capture** function. To recognize or identify objects, the Object Recognition **& Identification** subfunction may use a prior knowledge of the object provided by the Recognition & Identification Information Extraction subfunction of the World Storage function.

NOTE: Implementations of this subfunction can use machine learning approaches for the recognition of types of objects or the identification of unique objects.

5.2.5 Object Relocalization

This subfunction shall deliver the pose (position and orientation) of an object of the real world at present time without any prior knowledge of the pose of this object. It should use one or more subfunctions of the **World Capture** function and may use a prior knowledge of the object provided by the Relocalization Information Extraction subfunction of the World Storage function.

NOTE: The pose of the relocalized object can be referenced in the coordinate system of a sensor, of the device providing the AR experience, or of the world.

5.2.6 Object Tracking

This subfunction shall deliver the pose (position and orientation) of the object at present time based on information built from the previous capture (e.g. the pose of the object during the previous world capturing).

NOTE 1: The pose of the relocalized object can be referenced in the coordinate system of a sensor, of the device providing the AR experience, or of the world.

NOTE 2: An implementation of this subfunction can analyse the motion between the current data captured by the AR device and the data previously captured (e.g. based on optical flow or feature matching).

5.2.7 3D Mapping

This subfunction shall deliver a 3D representation of the real world based on a set of 3D features (e.g. points, edges). The features-based 3D representation can be constructed from data captured by one or more sensors. The pose and calibration parameters of these sensors should be known, and can be provided by other **World Analysis** or **World Capture** subfunctions. The methods used to construct the maps and also the density of the maps can differ depending on the type of sensors used.

NOTE: For image-based cameras, the 3D mapping can be implemented based on a triangulation of features which match for at least 2 frames knowing the pose and calibration parameters of the cameras which capture them. For depth sensors, the 3D mapping can be achieved by aligning and merging the current depth map defined in the world coordinate system (knowing the pose and calibration parameters of the sensor) with the current map built with the previous frame(s).

5.3 World Storage

5.3.1 Introduction

This function delivers information to other functions and subfunctions concerning the representation of the real world (e.g. information for relocalization, 3D object recognition and identification, AR authoring). Additionally, it receives information from other functions to update the representation of the world (e.g. 3D Mapping, 3D Object Recognition), it converts the world representation (Scene Meshing) or extracts part of this representation (Object 3D Segmentation) for the AR Authoring function.

5.3.2 World Representation

This subfunction shall create, update and provide an up-to-date global representation of the real world. This representation can result from the subfunction 3D Mapping of the World Analysis function, or from a representation of the real world produced by the Asset Preparation function and conveyed by the **AR Authoring** function. This global representation of the real world can be shared between various AR services.

NOTE: The representation of the world may vary and can include the following items:

- A fiducial marker: It is generally a marker which can be defined based on a black and white pattern that is fixed on an object for object relocalization or put in the scene for the AR device relocalization. These patterns can be square, circular, or more complex, and can generally be described by a Boolean vector. To deliver a correct localization of the AR device or of an object, the Object Relocalization subfunction should know the size of the 2D marker in a unit of length chosen for the overall AR system. This marker can be produced by the **Asset Preparation** function.

- A natural image marker: It is a 2D image that is put in the scene or fixed on an object. To deliver a correct localization of the AR device or of the object, the Object Relocalization subfunction should know the resolution of this image as well as the size of the 2D marker in a world unit of length chosen for the overall AR System. Generally, this marker can be represented by a set of 2D keypoints defined in pixel space (e.g. SIFT, SURF keypoint extractors). Each keypoint is associated with its descriptor, generally represented by an integer or float vector, aiming to characterize this keypoint in the most discriminating way based on neighborhood information in the image. This marker can be produced by the **Asset Preparation** function.

- A 3D structure: It is a set of 3D features (e.g. points, edges, meshes) representing the scene or an object. Each feature stores its 3D positions in the real world or in the object coordinate system as well as a descriptor, generally represented by an integer or float vector, aiming to characterize it in the most discriminating way in a 2D image. This 3D structure can be built by the **3D Mapping** subfunction or can be computed from a synthetic model (e.g. CAD or BIM 3D model) produced by the **Asset Preparation** function.

- A collection of shots: It is a structure allowing to approximate the pose of a captured query image using the pose of the most similar image retrieved from a database of shots. These shots are covering the space of the possible viewpoints from which the user will see the environment or objects. They generally use a bag of words representation to ease their retrieval.

- A trained model: It is a data structure (e.g. random forest, convolutional neural network) built during a learning stage. The Object Relocalization or **AR Device Relocalization** subfunctions can use this data structure to directly infer the pose of the AR device or of the object from the data delivered by the **World Capture** function.

- A combination of previous ones: An environment or an object can be represented by a combination of previous representations where each of these representations are defined in its own coordinate system, and where each of these coordinate systems is referenced in a common coordinate system of the world or of the object.

5.3.3 Relocalization Information Extraction

This subfunction shall extract from a global world representation the information required for relocalizing the device providing the AR experience or objects of the real world. A general task of this subfunction consists in extracting a portion of a global world representation and transform it into features used by the relocalization subfunctions of the **World Analysis** function. These features can be marker descriptions, 3D structure-based descriptions, 2D image-based descriptions, or machine learning-based descriptions.

5.3.4 Recognition & identification Information Extraction

This subfunction shall extract from a global world representation the information required to recognize or identify objects of the real world.

NOTE 1: It can consist of extracting information such as features describing the object from different viewpoints or audio listening points in order to recognize or identify them, and to format these features according to the specific implementation of the Object Recognition & Identification subfunction of the World Analysis function (e.g. features-based, appearance-based, or geometric fitting).

NOTE 2: A unique identifier can also be provided by e.g. a fiducial marker, a bar code or a RFID tag.

5.3.5 Object 3D Segmentation

This subfunction shall deliver a set of 3D objects which have been segmented from the global world representation.

NOTE: The process can consist of first classifying objects (identify the classes of objects seen by the system), then detecting objects (find the positions of the instances of these objects), segmenting objects (find the contours of these objects), and finally extracting them from the global world representation. The three first steps mainly use machine learning approaches.

5.3.6 Scene Meshing

This subfunction shall deliver a 3D mesh obtained from a portion of, or the complete world representation or from segmented objects.

NOTE: This subfunction can estimate the surfaces of the objects of the scene from a cloud of features such as points or edges more or less dense. This 3D mesh can be delivered to the AR Authoring function in order to render a 3D reconstruction of the real world or to manage occlusions between real and virtual contents by using this mesh as an occlusion mask.

5.4 Asset Preparation

5.4.1 Introduction

This function provides multimedia content to compose an AR scene. This content is mainly produced by external design tools. It can be interactive and can include audiovisual objects, 3D objects, behaviours attached to each object, and scenarios defining the different steps of an AR experience. This content shall be exported to the AR Authoring function to be composed together and attached relatively to the real world using the AR Scene Compositing subfunction. This content can be conveyed to the World Storage function to serve as objects of reference representing elements of the real world. Finally, this function can directly receive information from the Scene Management function to update the source of the content itself.

NOTE 1: Man-made multimedia content can be produced in a pre-processing step by a wide variety of design tools such as image, audio and video editors, CAD software, 3D computer graphics editors, PLM or BIM servers, and documentation systems.

NOTE 2: Content produced by the AR system itself representing the real world is handled by the World Storage function and data flow produced at runtime to change some properties of scene objects is handled by the External Application Support function.

5.4.2 Synthetic Content

Part of an AR scene can be synthetic content (e.g. 3D objects, 2D images, sounds, or haptic sensations) that are artificially generated.

This subfunction shall provide such objects in a representation that shall later be used for the creation of an AR scene.

5.4.3 AV content

Part of applications can be audio-visual objects as movies including audio or still pictures with accompanying speech. Single visual or auditory objects can also be part of a scene and can be overlaid on synthetic or audio-visual objects.

This subfunction shall provide such objects in a representation that shall later be used for the creation of an AR scene.

5.4.4 Object Behaviour

An AR scene should be interactive. This means that the elements, essentially the virtual content but also some real objects if they provide a control interface, can be modified in real-time according to user, content or environmental interactions. These modifications can be controlled by a pre-defined behaviour or by AI-based algorithms.

This subfunction shall deliver behaviours which can be attached to objects to modify them according to environmental, content or user interactions.

NOTE 1: One or multiple behaviours can be attached to an object and should only impact that object itself or the objects or user interfaces with which it interacts.

NOTE 2: Separating the behaviours from the objects themselves should offer more reusability as the same behaviour can be applied to several objects. For instance, a behaviour to move an object according to user interactions can be applied to any movable object of the scene, the same can hold for a behaviour that highlights the object when it is selected.

5.4.5 Scenario

A scenario consists of several objects and a description of their availabilities and dependencies, not only between those objects but also in relation to user, content or environmental interactions.

This subfunction shall deliver a description of a scenario in a representation that shall later be used for the creation of an AR scene.

NOTE 1: The scenario can be implemented with a state machine defining the rules on how the state of scene objects will evolve over time according to user, content or environmental interactions.

NOTE 2: Separating the scenario from the behaviours and objects should offer more reusability, as the same object with the same behaviours attached to it can be instantiated several time in the same scenario (e.g. a 3D object of a screw with a behaviour showing an animated screwdriver when activated by the scenario).

NOTE 3: The scenario can be imported from procedure design tools.

5.4.6 Report Evaluation

This subfunction processes the reports received in real time from the **Scene Management** function. The result of this evaluation may be transmitted to the other subfunctions of the **Asset Preparation** function for updating content used to compose the AR scene. Also, these results could be sent to external systems (e.g. a PLM tool or a CAD editor).

5.5 External Application Support

This function shall handle real-time communication and data exchange between the AR implementation and an external system. An external system may provide data at runtime which can modify the AR scene. It may also receive data or commands from the AR implementation to control the external system. The potential types of external systems can be manifold. It ranges from a simple sensor measuring a certain data value up to a complex system with multiple components. In the present document the required communication and data exchange functionalities are subsumed under the term External Application Support.

5.6 AR Authoring

5.6.1 Introduction

This function can help to prepare an AR scene by using assets or models built by the **World Representation** to prepare an AR scene which is composed of several scenic objects with specific behaviours and their inter-relations positioned relatively to the real world or user space. This function provides the AR scene to the **Scene Management** function and content to the **World Representation** function.

5.6.2 Content Conversion

This subfunction shall convert content exported from the Asset Preparation function to a representation understandable by the Scene Management function.

NOTE: Many different formats exist to represent a 3D scene composed of structured 3D objects with associated metadata.

5.6.3 Content Optimization

This subfunction shall optimize the content to be sent to the Scene Management function for rendering. Depending on the capabilities and performance of the 3D Rendering function, the Scene Management function can control the AR Authoring function by requesting for instance to simplify the complexity of the content.

NOTE: Optimization can consist of reducing the number of polygons of 3D objects with a mesh dissemination approach, or reducing the size of textures, or removing unnecessary 3D objects.

5.6.4 AR Scene Compositing

This subfunction shall build an AR scene. This can be built by attaching object behaviours to AV or synthetic contents, then by injecting these objects with their behaviours in a scenario. This subfunction should also position some of these objects relatively to anchors provided by the World Storage function to ensure that these interactive objects are perceived well positioned by the user relatively to the real world. These objects can also be positioned relatively to the user space.

AR anchors, representing reference coordinate systems related to an element of the real world, shall be defined by the World Storage function.

NOTE: Objects can be positioned relatively to the user. In this case, if the user moves the head, objects will remain static in the user's field of view or acoustic space.

5.6.5 Content Packaging

This subfunction shall format the whole or a part of the AR scene according to a desired packaging format. Dynamic packaging can also be supported in such a way that the output of the packager offers differently encoded versions of an object accompanied by switching points that are aligned across different representations of the same object (e.g. to take into account variable network conditions or rendering performances).

5.6.6 Content Hosting

This subfunction shall make available the AR scene including virtual content and their behaviour for delivery to the Scene Management function.

NOTE: The Content Hosting subfunction may be realized by simple web servers, as part of an origin cluster, or operated by a distributed Content Delivery Network system.

5.7 User Interactions

5.7.1 Introduction

This function delivers information produced by user interaction devices. It can address different kind of sensors relative to user interactions such as tactile surfaces, 3D gesture recognition, vocal interface, gaze tracking, or biometric sensors.

5.7.2 3D Gesture

This subfunction shall provide information related to the user's 3D gestures. This may involve the movement of hands, arms, and legs. Two kinds of information may be generated by this subfunction. Firstly, this subfunction can provide a 3D gesture that has been recognized (e.g. a swipe in the air, a specific posture of the user's body) and can be considered as a command. Secondly, this subfunction can provide in real-time the skeleton pose of the user, i.e. the position and/or orientation of each or certain joints of the user's body.

NOTE: The pose of the user's skeleton can be used, for example, to animate an avatar representing the user, or to detect when a part of the user's body touches virtual or real elements.

5.7.3 Tactile

This subfunction shall provide information related to the contact of the user's fingers on a tactile screen or on any surface. Two kinds of information may be generated by this subfunction. Firstly, this subfunction can provide a 2D gesture that is known to the application (e.g. a tap, a double tap, a press, a 2D swipe, a 2D pinch in or out, or a recognized character) and can be considered as a command. Secondly, this subfunction can provide in real-time the 2D contact points with the surface, that can be used, for example, to move a virtual object relatively to the real environment.

5.7.4 Gaze

This subfunction shall provide information relative to the user's gaze. Two kinds of information may be generated by this subfunction. By analysing the movement of the user's eyes, this subfunction can detect their viewing direction relative to the head of the user (e.g. a ray starting from a point positioned between the two eyes and pointing in a given direction), and/or the 3D focus point of the user providing depth information concerning the user's gaze.

NOTE: To provide gaze information related to the real world, it will require to be combined with a head tracking system.

5.7.5 Vocal

This subfunction shall recognize and transform a spoken language into a textual representation or a set of commands to the **Scene Management** function.

NOTE: The command can consist of a single command word that triggers an action ("Status"), or a command composed of several words that influences an object selected by an associated word or by another interaction method ("Switch on lamp one" or select the target lamp by the gaze and say "switch on"), or a phrase whose meaning is decided according to a known context "Put that there".

5.7.6 Biometric

This subfunction shall provide information based on physiological signals emitted by the user's body.

NOTE: These physiological information can be related to various signals such as heart rate and amplitude (electrocardiogram), the electrodermal activity providing information about the sweating of the user, the electrical activity of the brain of the user (electroencephalography), the electrical activity produced by skeletal muscles (electromyography), or the blood volume changes in the microvascular bed of tissue (photoplethysmogram).

5.8 Scene Management

5.8.1 Introduction

This function manages at runtime an AR scene representation related to a specific user. It is a core function connected to mainly all other functions of the framework. The AR scene representation shall be sent to the 3D Rendering function. This representation could be updated according to inputs coming from user interactions (**World Analysis** and **User Interactions** functions). It could also be updated according to content interactions defined by their behaviours or a scenario, or new content sent by either the AR Authoring function, or the World Capture function, or both. This function could also send some reports to the **Asset Preparation** function.

5.8.2 Interaction Technique

This subfunction shall provide interaction tasks deduced from signals sent by the User Interactions function.

NOTE 1: Four categories of universal interaction tasks should be used for AR applications: selection (specifying one or more objects from a set), manipulation (modifying object properties, most often its position and orientation), system control (e.g. changing the system state or the mode of interaction) and symbolic input (e.g. communicating text, numbers, and other symbols or marks to the system).

NOTE 2: For an AR system, the navigation task should be performed by the World Analysis function, as the user navigates at scale 1:1 according to the real space.

NOTE 3: This interaction technique can provide user feedback related to the interaction task (e.g. virtual hands, virtual ray, sounds and haptic feedback).

5.8.3 Virtual Scene Update

This subfunction shall maintain an up-to-date representation of the scene rendered to the user. This representation of the scene may be updated according to connected functions and subfunctions such as:

- World Capture (e.g. updating the background texture with an image captured by a camera for video see-through displays).
- World Analysis (e.g. updating the pose of the virtual viewpoint according to the pose estimation of the AR device or updating the position and orientation of a virtual object according to the pose estimation of a real object).
- Interaction techniques (e.g. updating the properties of an object when manipulating it according to the behaviour of this object).
- 3D Rendering (e.g. provisioning the scene for rendering or updating the complexity of the scene according to 3D rendering performances).
- AR Authoring (e.g. adding new objects to the scene or updating the occlusion mask with the 3D reconstruction of the real space coming from the World Storage function).
- External Application Support (e.g. injecting a flow of data on a property of an object of the scene).
- Asset Preparation (e.g. reporting property updates or feedbacks to an external application from which the content is originally coming, such as a PLM or a BIM).
- Transmission (e.g. synchronizing scene graph with the scene graph of a distant AR system for multi-users applications).

NOTE: Implementations of this internal representation should be structured through a 3D scene graph updated by a loop synchronized with the 3D Rendering function.

5.8.4 Content Unpackaging

This subfunction shall extract the content of an AR scene from a package and present it to the Virtual Scene Update subfunction.

5.8.5 AR Experience Reporting

This subfunction shall deliver reports related to the AR Experience back to the AR Authoring or Asset Preparation functions. This may cover aspects as the behaviour and the quality of experience of the presented scene, information related to the user's interactions or to the status of the system itself. The receivers of these reports can take them into account for improving the AR experience or can convey them to external systems.

5.9 3D Rendering

5.9.1 Introduction

This function fully or partially renders in real-time the current AR scene representation to transmit the result to the Rendering Adaptation function.

5.9.2 Video

This subfunction shall generate one or several images of an AR scene representation showing the scene from one or several viewpoints.

5.9.3 Audio

This subfunction shall generate the sound environment for one or several 3D listening points produced by one or several audio sources positioned in the 3D space (diegetic sound) or attached to the listening point (extradiegetic sound).

5.9.4 Haptic

This subfunction shall generate the forces, vibrations, textures or motions to simulate different haptic feedbacks to a user when interacting with the scene.

5.10 Rendering Adaptation

5.10.1 Introduction

This function adapts the images, sounds, forces and vibrations produced by the 3D Rendering function to the specific rendering devices such as video see-through displays, optical see-through displays, projection-based displays, audio speakers and haptic systems.

5.10.2 Video see-through

This subfunction shall adapt the images produced by the 3D Rendering function to the specific characteristics of a video see-through display.

NOTE 1: A video see-through device is a system that captures the real environment with one or more cameras, and displays the captured image to the user with one or more screens. The device can be hand-held (e.g. a tablet or a smartphone) or can be worn on the head of the user (an AR headset).

NOTE 2: Generally, no adaptation to the image is required for hand-held video see-through devices, as 3D rendering engines are normally designed for single flat displays.

NOTE 3: A video see-through AR headset can consist of two opaque screens and two cameras located on the front face of the headset and ideally positioned in the axis of the user's eyes. The video of each camera is displayed on the background of the AR scene. The resulting rendered image mixing these videos and virtual content should be adapted to the characteristics of the video see-through headset (e.g. radial distortion of the optical video see-through system, calibration parameters of the cameras, positions of the cameras according to the screens and positions of the user's eyes in relation to the screens).

5.10.3 Optical see-through

This subfunction shall adapt the images produced by the 3D Rendering function to the specific characteristics of an optical see-through display.

NOTE: An optical see-through device can consist of one or two transparent screens for the perception of the real world. As optical see-through systems should combine a set of complex optical elements, the images resulting from the 3D Rendering function should be adapted by transforming them according to the optical see-through display characteristics (e.g. radial distortion of the optical see-through display and positions of the user's eyes in relation to the screens).

5.10.4 Projection-based

This subfunction shall adapt the images produced by the 3D Rendering function to the specific characteristics of a projection-based display.

NOTE: Projection-based systems can consist of one or several video projectors that will display the AR scene on flat or irregular surfaces. As a result, the images resulting from the **Video** subfunction of the **3D Rendering** function should be transformed according to firstly the projective system characteristics (e.g. focal, position and orientation, gamma correction, blending model) and secondly the surface on which the AR scene is projected (warping model relative to the relief of the surface).

5.10.5 Audio

This subfunction shall adapt audio produced by the **3D Rendering** function to the specific characteristics of a sound system.

NOTE: Such a system consists of at least one speaker. Stereo effects can be achieved by mapping the acoustic sources of an AR scene to the stereo base spanned by two loudspeakers. By positioning additional loudspeakers, spatial sound effects can be reproduced with more fidelity.

5.10.6 Haptics

This subfunction shall adapt haptic information produced by the 3D Rendering function to the specific characteristics of the reproductive mechanism. So, the user can receive an AR-related information in the form of a felt sensation on some part of the body.

5.11 Transmission

5.11.1 Introduction

This function conveys data transmitted between functions shown in Figure 2 and/or external systems while considering the characteristics of the transmission path. This function addresses most of the reference points mentioned in clause 6. This function can be an enabler for establishing connectivity via different technologies over a wired or wireless network.

The communications between functions and external systems cause hardware and software requirements for the AR application. For example, the communication could be done via Wi-Fi™, 3G, 4G or 5G.

5.11.2 Security

This subfunction shall encrypt and decrypt data conveyed between different functions, subfunctions and/or external systems. This subfunction shall include a conditional access management system which should deliver encryption and decryption keys or certificates in line with existing usage rules guiding the transmission function.

5.11.3 Communications

This subfunction shall encode and decode data conveyed between different functions, subfunctions and/or external systems. It aims at reducing the amount of data to efficiently use the available bandwidth. Coding schema should be relevant to the type of data which is being transmitted through the transmission function (e.g. image, video, audio, 3D content, behaviour, AR scenario, map, or runtime report).

NOTE: A single AR content may be transformed into several different encoded representations to match delivery conditions.

5.11.4 Service Conditions

This subfunction shall provide service-related parameters of the used transmission path in collecting the outcome of performance and making it available for runtime reports. In case of mismatch between network conditions and recommended scene quality parameters, the Scene Management function can request to adapt the AR scene to the existing service parameters.

6 Reference Points of the Functional Architecture

6.1 Introduction

The architecture in Figure 2 shows the functional elements for the processing and the deployment of AR implementations and the reference points between them.

6.2 Reference point "Sensors for World Analysis AR1"

Reference point AR1 between functions World Capture and World Analysis shall offer support for the analysis of the real world by providing sensor-related information, namely sensors' capabilities and characteristics, their settings and controls, and the delivery of the data they capture. This reference point should support the following functionalities:

- Access to sensor capabilities and characteristics:
    - Type of sensor (e.g. RGB camera, RGB-D camera, accelerometer, magnetometer, microphone, or GPS).
    - Identifier of the sensor provided by the **World Capture** function.
    - Number of devices of similar type.
    - Measuring characteristics (e.g. range and resolution, acquisition frequency, conditions of use, or coverage).
    - Information about the extrinsic, intrinsic and distortion parameters of the sensors.
- Control of the sensor:
    - Initializing and shutting-down sensors.
    - Starting and stopping the access to sensors.
    - Setting of identifiers as used by the **World Analysis** function.
    - Setting of measurement characteristics (e.g. ranges, systems of units, resolutions, or acquisition frequency).
    - Switching between sensors.
    - Requesting conditions for data delivery to the AR implementation (e.g. event-based, temporal, or synchronized with other sensors).
- Delivery of captured data:
    - Sensor data as e.g. raw data or aggregated or converted data.
    - The ID of the sensor which has captured the data.
    - Metadata associated to data captured by the sensor.

NOTE 1: Extrinsic parameters can provide the pose of the sensor relatively to reference coordinate systems related to the AR device, to another sensor, or to the real world.

NOTE 2: The metadata associated to the captured data can be an actual measurement unit or a time stamp for representing time of data acquisition (essential to merge data supplied by multiple sensors).

6.3 Reference point "Sensor Data for Scene Management" AR2

Reference point AR2 between functions World Capture and Scene Management shall offer support for managing at the AR scene representation related to a specific user by providing sensor-related information, namely sensors' capabilities and characteristics, their settings and controls, and the delivery of the data they capture. This reference point should support the following functionalities:

- Access to sensor capabilities and characteristics:
    - Type of sensor (e.g. RGB camera, RGB-D camera, accelerometer, magnetometer, microphone, or GPS).
    - Identifier of the sensor provided by the **World Capture** function.
    - Number of devices of similar type.

  - Measuring characteristics (e.g. range and resolution, acquisition frequency, conditions of use, or coverage).
  - Information about the extrinsic, intrinsic and distortion parameters of the sensors.

- Control of the sensor:
  - Initializing and shutting-down sensors.
  - Starting and stopping the access to sensors.
  - Setting of identifiers as used by the **Scene Management** function.
  - Setting of measurement characteristics (e.g. ranges, systems of units, resolutions, or acquisition frequency).
  - Switching between sensors.
  - Requesting conditions for data delivery to the AR implementation (e.g. event-based, temporal, or synchronized with other sensors).

- Delivery of captured data:
  - Sensor data as e.g. raw data or aggregated or converted data.
  - The ID of the sensor which has captured the data.
  - Metadata associated to data captured by the sensor.

NOTE 1: Extrinsic parameters can provide the pose of the sensor relatively to reference coordinate systems related to the AR device, to another sensor, or to the real world.

NOTE 2: The metadata associated to the captured data can be an actual measurement unit or a time stamp for representing time of data acquisition (essential to merge data supplied by multiple sensors).

NOTE 3: This reference point can be used to update video background information as part of the AR scene for video see-through AR systems.

NOTE 4: This reference point can be used to update the audio background in case of an implementation offering augmented audio functions.

6.4 Reference Point "External Communications" AR3

This reference point between functions Scene Management and External Application Support shall control communications with external systems and conveying associated runtime data that is used for the actual implementation of the AR application. The functionalities of this reference point are not further specified in the present document.

NOTE: Runtime data may be a command coming from the **Scene Management** function to control at runtime an external application, or a stream of data provided by the **External Application Support** function which is injected at runtime into the AR scene.

6.5 Reference Point "User Interactivity" AR4

This reference point between functions User Interactions and Scene Management shall provide information about user's interactions captured by input devices to steer, control or influence the behaviour of the AR scene. This reference point should support the following functionalities:

- Delivery of signals captured by the input devices (gestures, tactile signals, body pose, vocals, gaze, biometric).
- Delivery of the state of the input devices (activated, detected, confidence in the signals).
- Provisioning of input device characteristics and capabilities.
- Setting of input device characteristics.

NOTE 1: The signals captured by input devices can be represented by the values returned by the user interface according to the number of degrees of freedom (DoF) they can handle. The spaces of the degrees of freedom can be discrete or continuous, limited or unlimited. For instance, a button can generate a binary or 0 DoF action, a vocal command can generate a 1 DoF action in a limited discrete space, a throttle can generate as 1 DoF action in a limited continuous space, a joystick can generate a 2 DoF action in a continuous limited space, a mouse can generate a 2 DoF action in a continuous unlimited space, a VR or AR controller tracked in the air can generate a 6 DoF action in a continuous and limited space (the tracking area), or a hand or body tracker can generate in a continuous and limited space an *n* DoF action (*n* being the number of joints that can be tracked by the system).

NOTE 2: An input device can be characterized by its type (e.g. gesture, tactile, vocals, gaze, heartbeat, blood pressure), the number of degrees of freedom of its signals, a description of the space for each degree of freedom (continuous or discrete and limited or not), or the unit of the values returned by the user interface. Additionally, an input device can be characterized by a description of the element of the real world to which it should be attached (e.g. the user's head if mounted on a helmet, the right or left hand if handled by the user, the part of the body to which it is attached or a tool or any object to which it is fixed such as a drill or a gun).

6.6 Reference Point "Rendered Scene" AR5

This reference point between functions 3D Rendering and Rendering Adaptation shall convey the representation of the rendered AR scene in a media-related way for real-time purposes. This reference point should support the following functionalities:

- Delivery of a representation of audio and visual objects in implementation-specific formats.
- Delivery of a representation of haptic events in implementation-specific formats.
- Provisioning of rendering device characteristics and capabilities.
- Setting of rendering device characteristics.

NOTE: For visual content, the representation of a sequence of images should consist in 2D images generated by the rendering engine which can benefit from one or more graphic processing units. For audio content, the representation should consist of audio signals based on a multichannel surround formats such as Stereo, Quad, 5.1, 7.1, 22.2, ambisonics, HOA (High Order Ambisonics) and other formats.

6.7 Reference Point "Rendering Performances" AR6

This reference point between functions 3D Rendering and Scene Management shall give feedback about the quality of rendering to optimize the scene representation. This reference point should support the following functionalities:

- Reporting on the performance of the 3D Rendering function (e.g. frequency or latency).

6.8 Reference Point "Scene Representation" AR7

This reference point between functions Scene Management and 3D Rendering shall convey the description of the AR scene based on all objects that make the scene. This reference point should support the following functionalities:

- Initialization of the AR scene.
- Addition and deletion of single objects of the AR scene.
- Change of properties of objects (including the pose of a virtual camera representing the viewpoint of the user).

6.9 Reference Point "Pose" AR8

This reference point between functions World Analysis and Scene Management shall provide information about the position and the orientation of real world objects or of the AR device hosting the AR implementation. This reference point should support the following functionalities:

- Provisioning of reference coordinate system.
- Provisioning of the estimated position and orientation of the device that is hosting the AR implementation relatively to the reference coordinate system.
- Provisioning of the estimated position and orientation of real world objects relatively to the reference coordinate system.
- Delivery of metadata associated to the pose.

NOTE 1: The reference coordinate system, also called anchor, can be relative to different elements of the real world (e.g. the real world on its own, an object, or the device hosting the AR implementation).

NOTE 2: Metadata associated to the pose can be e.g. actual measurement unit, time stamp for representing time of captured data, or metric concerning the confidence of the estimated pose.

6.10 Reference Point "Recognized or Identified Object" AR9

This reference point between functions World Analysis and Scene Management shall convey information about objects of the real world that have been recognized or identified. This reference point should support the following functionalities:

- Provisioning of recognized or identified objects in an application-specific data format.
- Provisioning of information about the state or the behaviour of recognized or identified objects within the real world.

6.11 Reference point "3D Map" AR10

This reference point between functions World Analysis and World Storage shall convey 3D information about the geometry of the real world. This reference point should support the following functionalities:

- Provisioning of information about the type of model used for the 3D representation of the real world.
- Provisioning of coordinate reference systems and their scaling.
- Provisioning of information about data formats used for the 3D representation of the real world.
- Provisioning of 3D information and metadata about the geometry of the real world.

NOTE 1: The 3D representation can be based on different types of models such as a cloud of 3D features (e.g. 3D points and 3D edges), images, trained models (e.g. Random Forests or Convolutional Neural networks).

NOTE 2: The metadata associated to 3D information can be e.g. a descriptor, a colour or a normal to the surface associated to a 3D feature.

6.12 Reference Point "Relocalization Information" AR11

This reference point between functions World Analysis and World Storage shall convey information regarding the relocalization of an element of the real environment. This reference point should support the following functionalities:

- Provisioning of information about the type of model used for the relocalization information.
- Provisioning of coordinate systems and their scaling.

- Provisioning of information about data formats used for the relocalization information.
- Provisioning of relocalization information according to criteria related to the **World Analysis** function.

NOTE 1: Elements of the real world that can be relocalized are any physical object including the AR device itself, a user, a virtual content displayed by a physical object (e.g. a video on a screen), etc.

NOTE 2: Relocalization information can use different types of models e.g. an image-based representation such as visual bag of words, a 3D-based representation such as a point cloud with associated descriptors or a trained model such as Random Forests or Convolutional Networks.

NOTE 3: Criteria for provisioning of relocalization information can be e.g. a given region or a given environmental context related to the AR scenario such as for a day or night use.

NOTE 4: The type of model for the relocalization information provided by the **Relocalization Information Extraction** subfunction should be compliant with the one used by the **AR Device or Object Relocalization** functions. For instance, if the **Relocalization Information Extraction** subfunction provides a 3D-based representation model with a descriptor of type A attached to each 3D point, and if the **AR Device or Object Relocalization** functions cannot interpret this type of descriptor, they will not be able to perform the relocalization.

6.13 Reference Point "Recognition & Identification Information" AR12

This reference point between functions World Storage and World Analysis shall convey information about the recognition and identification of objects in the real world as part of the global world representation. This reference point should support the following functionalities:

- Provisioning of information about the type of model used for the recognition or identification information (e.g. an image-based representation such as visual bag of words, a 3D-based representation such as a point cloud with associated descriptors or a trained model such as Random Forests or Convolutional Networks).
- Provisioning of information about data formats used for the recognition or identification information.
- Provisioning of recognition or identification information according to the criteria related to the AR scenario (e.g. a subset of objects to recognize, a given environmental context related to the AR scenario such as for a day or night use).

NOTE 1: This subfunction can address both recognition and identification tasks, all depends of the labelling of the input data.

NOTE 2: Recognition or identification information can use an image-based representation such as visual bag of words, a 3D-based representation such as a point cloud with associated descriptors or a trained model such as Random Forests or Convolutional Networks, etc.

NOTE 3: Criteria for provisioning of recognition or identification information can be e.g. subset of objects to recognize or identify, or a given environmental context related to the AR scenario such as for a day or night use.

6.14 Reference Point "Scene Objects" AR13

This reference point between functions AR Authoring and Scene Management shall convey the objects of an AR scene or updates related to objects of an AR scene and information about their relationship, either at the request of the Scene Management function or at the decision of the AR Authoring function. This reference point should support the following functionalities:

- Requesting objects of an AR scene.
- Requesting a scene description of an AR scene.
- Requesting an update of AR scene objects.

- Updating properties of AR scene objects.
- Updating the AR scene description.
- Adding or deleting features of AR scene objects.
- Delivery of objects of an AR scene.
- Delivery of a scene description for an AR scene.
- Provisioning of information about the capabilities of the objects and their possible relationship.

6.15 Reference Point "AR Session Reports" AR14

This reference point between functions Scene Management and Asset Preparation shall report information captured at runtime during an AR session. This information can concern the behaviour and the quality of the AR experience of the presented AR scene, the user's interactions, or the status of the system itself. It is reported directly to the **Asset Preparation** function. This reference point may support the following functionalities:

- Reporting information about AR scene updates.
- Reporting information about user's interactions.
- Reporting information about the quality of the AR experience.
- Reporting information about the status of the system.

NOTE: This reference point can be used to report some updates that have to be validated before being applied to the sources of assets (e.g. a PLM or BIM server). For instance, annotations with snapshots captured during a maintenance or a quality control process can be reported through this reference point to a centralized asset management system. These reports can be validated by a design team and can be applied to the assets for future AR sessions.

6.16 Reference Point "3D Objects of World" AR15

This reference point between functions World Storage and AR authoring shall provide 3D representation of objects (the representation of the whole real world can be considered as a set of objects) reconstructed from observations of the real world. This 3D representation may be added to an AR scene in the **AR Authoring** function. This reference point should support the following functionalities:

- Provisioning of information about the type of model used for the 3D representation of objects.
- Provisioning of information about data formats used for the 3D representation of objects.
- Delivery of 3D objects.
- Provisioning of metadata about the 3D objects.

NOTE: Metadata other than the type of model or data formats could include keywords describing the 3D objects, provenance, its quality, or a timestamp.

6.17 Reference Point "World Anchors" AR16

This reference point between functions World Storage and AR Authoring shall provide the different anchors (coordinate systems attached to several elements of the real world) in order to attach virtual content to them. This reference point should support the following functionalities:

- Provisioning of 3D reference coordinate systems (anchors).
- Provisioning the 3D reference coordinate systems of higher level to which these reference coordinate systems are attached.

- Information on the kind of coordinate system used for world anchors (e.g. geodetic or cartesian).
- Information on the real world elements to which these reference coordinate systems are related.

NOTE: World anchors can be hierarchically organized. For instance, a world anchor related to a tool can be attached to a world anchor related to a machine which can be attached to a world anchor related to a factory, which can be attached to a world anchor related to the earth.

6.18 Reference Point "Reference Objects" AR17

This reference point between functions AR Authoring and World Storage shall provide an object as representation of an element of the real world for relocalization, recognition or tracking purposes. This reference point should support the following functionalities:

- Provisioning of new objects representing a physical element of the real world for an AR scene.
- Provisioning of both pose and scale of objects in relation to a coordinate system of the real world.
- Updating of properties of objects used for the representation of the real world.
- Deletion of objects from the representation of the real world.

NOTE: A representation of a physical element of the real world can be 2D markers (e.g. fiducial or based on a natural image), 3D models (e.g. CAD or BIM model), or videos.

6.19 Reference Point "Content export" AR18

This reference point between functions Asset Preparation and AR Authoring shall provide all assets required to author an AR scene (synthetic or audio-visual content, object behaviours, scenario). This reference point should support the following functionalities:

- Requesting synthetic or audio-visual content.
- Requesting object behaviours.
- Requesting scenarios.
- Delivery of synthetic or audio-visual content.
- Delivery of object behaviours.
- Delivery of scenarios.
- Provisioning of information about content formats and presentation details.

7 Use case implementation samples (informative)

7.1 Try before buying with AR

7.1.1 Use case description

In retail, AR is a tool of choice to try furniture in situ before buying it. Figure 3 shows an office augmented with a new chair. The AR scene is triggered by a 2D marker.

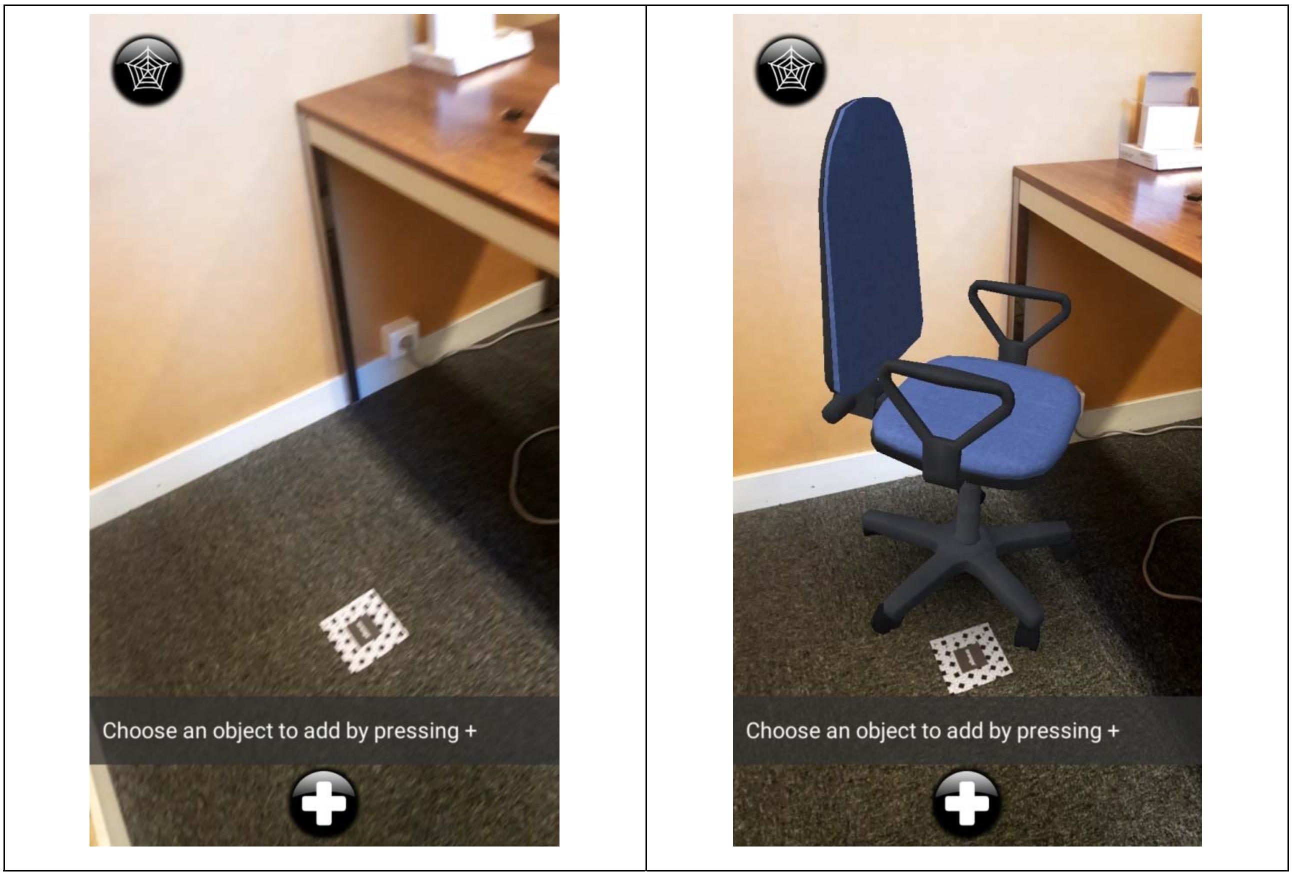

Figure 3: An office augmented with a new chair

A simple way to achieve this consists in:

- During preparation:
  - Creating a marker based on a 2D image.
  - Placing a marker at a defined 3D position/orientation in the real world.
  - Getting the 3D model of the chair.
  - Placing the 3D model relatively to the coordinate system of the marker in the real world.
- During the AR experience:
  - Relocalize the AR device relatively to the marker.
  - Displaying the 3D model of the new chair on the device.
  - Perform tracking operations to keep synchronized while moving.

7.1.2 Use case implementation

The workflow diagram in Figure 4 describes an example of the implementation of the AR framework for the use case "Try before buying with AR".

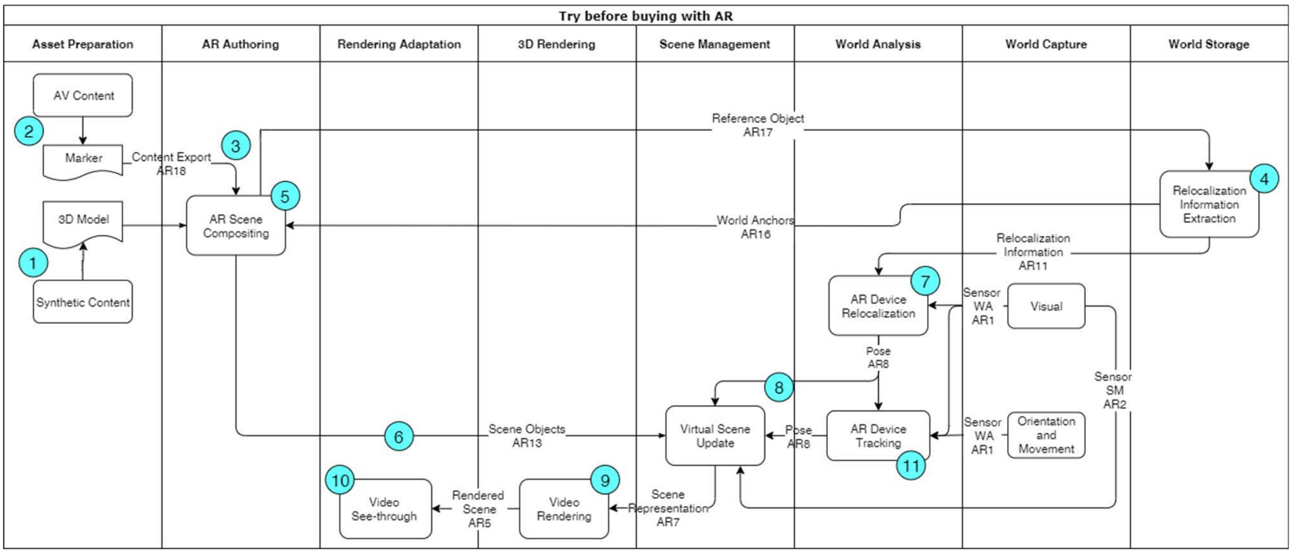

Figure 4: Workflow diagram of the subfunctions implemented for the use case "Try before buying with AR"

1) With a 3D computer graphics software a 3D graphic designer produces a 3D model of the chair with textures and materials (subfunction Asset Preparation/Synthetic Content) that will be displayed in augmentation. This 3D model is defined in a given unit (e.g. meter).

2) With a 2D computer graphic software, a 2D designer produces a 2D natural image marker with the subfunction Asset Preparation/AV Content that will be sent to the subfunction AR Authoring/AR Scene Compositing.

3) The AR service developer exports the image to be used as a 2D natural marker and adds it in the list of potential reference objects with the subfunction AR Authoring/AR Scene Compositing.

4) The 2D natural image marker is uploaded to a server to extract relocalization features (subfunction World Storage/Relocalization Information Extraction). These relocalization features are defined in the coordinate system related to the image marker and is sent back to the subfunction AR Authoring/AR Scene Compositing as a World Anchor (for example relative to the center of the natural image marker, with x-axis along the width of the marker, y-axis along the height of the marker, z-axis orthogonal to the marker, and with a meter-unit).

5) The 3D model of the chair is positioned according to the return world anchor related to the natural image marker (subfunction AR Authoring/AR Scene Compositing).

6) Now, a visitor can launch its AR application on a tablet. First, the 3D model of the building will be loaded in a scene graph.

7) The camera will be launched (subfunction World Capture/Visual) and the first image captured by the camera will be sent to the subfunction World Analysis/AR Device Relocalization. This last subfunction will use the relocalization information provided by the subfunction World Storage/Relocalization Information Extraction to estimate the first pose of the tablet without any knowledge of its original position.

8) This first pose of the tablet will be sent to the subfunction Scene Management/Virtual Scene Update to update the position of the virtual camera in the scene graph, and the subfunction World Capture/Visual will transmit the captured image to update the background image in the scene graph.

9) The rendering engine will traverse the scene graph thanks to the subfunction 3D Rendering/Video to create an image of the 3D model of the chair seen from the virtual camera (corresponding to the viewpoint captured by the camera of the tablet) with in background the image captured by the camera.

10) This image coming from the 3D renderer will be adapted by the subfunction Rendering Adaptation/Video see-through to be displayed on the screen of the tablet (Actually, no real adaptation is needed on a tablet as there is no need to distort the image for dedicated lenses).

11) The visitor moves in the room, and each time an image is captured by the camera or data are captured by the IMU (subfunctions of World Capture function), an update of the pose of the AR device is done using the subfunction World Analysis/AR Device Tracking that will use the pose of the AR device at the previous frame to estimate the pose of the AR device at the current frame. This new pose of the AR device will be sent to the subfunction Scene Management/Virtual Scene update, and the steps 8, 9 and 10 will loop until the application closed.

7.2 Maintenance Support

7.2.1 Use case description

In today's industry the technical complexity of workflows is increasing and although the degree of automation rises, human skills and dexterity are still needed for the delivery of a product and the deployment of a service. Especially in the area of telecommunications, outdoor activities of employees are still part of the daily job. The example described in the present clause deals with the support for wiring in a street cabinet, a scenario that is depicted in Figure 5.

Figure 5: Wiring in a street cabinet

The scenario can be split into two working phases:

- Preparatory steps:
    - Creating a unique marker for the street cabinet.
    - Placing this marker at a known position inside the cabinet.
    - Using this marker as an identifier for a 3D model of the cabinet.
    - Aligning the 3D model's coordinate system with the position of the marker inside the cabinet.
- Work steps:
    - Marking in the 3D model the pins to be connected.
    - Tracking the pins.
    - Vocal confirmation when pins are connected.

7.2.2 Use case implementation

The workflow diagram in Figure 6 describes an example of the implementation of the AR framework for the use case "Maintenance support".

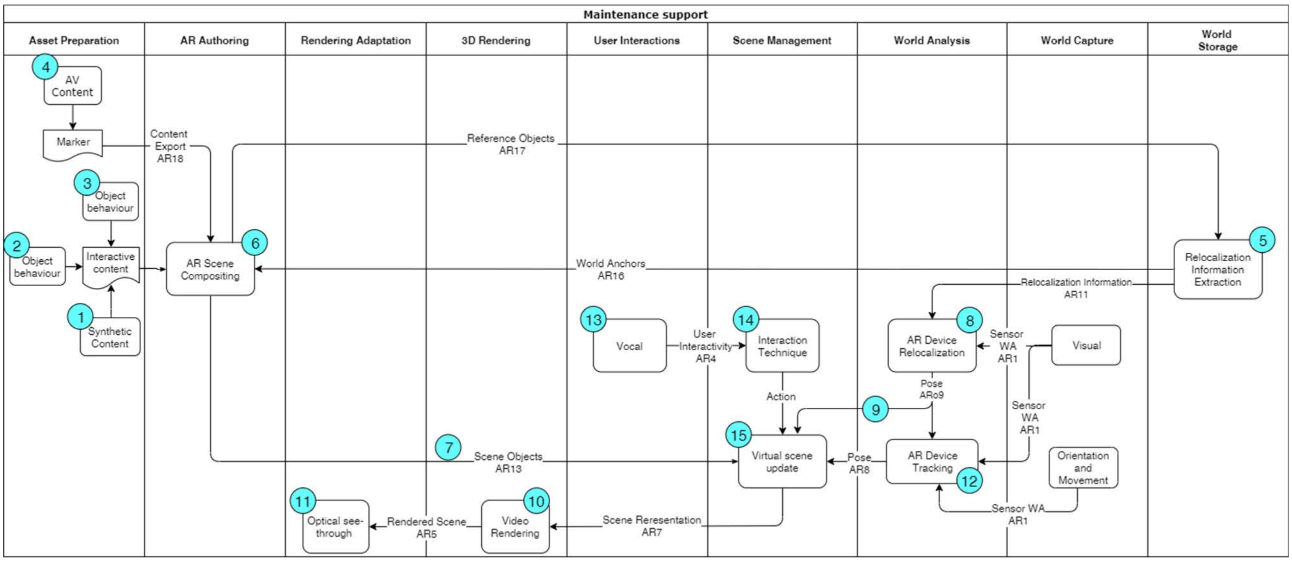

Figure 6: Workflow diagram for the use case "Maintenance support"

1) With a graphics software a graphic designer produces a model of the circuit board with connection strips and pins to establish connectivity between the telephone exchange and the telephone wall jack in the customer's home. The lay-out of this model reflects the circuit board actually used in this cabinet (subfunction Asset Preparation/Synthetic Content) and will be displayed in augmented reality. Two pins each to be connected are marked in color. The model is defined in a given measuring unit.

2) With an animation software or a 3D game editor, a 3D animator will design the behaviour of each pin showing technicians which pins to connect.

3) With a dedicated design tool to create scenarios, an expert defines the different steps of the wiring procedure consisting here of a series of pins that will have to be connected. The scenario defines that it will move to the next pair of pins as soon as it receives a "Next Step" action event.

4) With a 2D computer graphic software, a 2D image (subfunction Asset Preparation/AV Content) is produced that will be placed in the street cabinet for unambiguous identification of this cabinet. Its position in the cabinet is clearly defined and thus will later also be used for mapping the augmentation with reality.

5) The AR service developer exports the image to be used as a 2D natural marker and add it in the list of potential reference objects with the subfunction AR Authoring/AR Scene Compositing.

6) The 2D natural image marker is uploaded to a server to extract relocalization features (subfunction World Storage/Relocalization Information Extraction). These features are defined in a given coordinate system which is sent back to the subfunction AR Authoring/AR Scene Compositing as a world anchor (for example relative to the center of the natural image marker, with x-axis along the width of the marker, y-axis along the height of the marker).

7) The model of the circuit board with associated behaviours and scenario is positioned according to the world anchor related to the natural image marker (subfunction AR Authoring/AR Scene Compositing).

8) The technician launches the AR support application on its headset and the model of the circuit board with associated animations and scenario will be loaded (subfunction Scene Management/Virtual scene update*)*. The virtual scene is initialized with the first step of the scenario (the first pair of pins to be connected is activated, following ones are deactivated).

9) The built-in camera of the AR headset is launched (subfunction World Capture/Visual) and the first image being captured by the camera is sent to the subfunction World Analysis/AR Device Relocalization. This subfunction uses the relocalization information provided by the subfunction World Storage/Relocalization Information Extraction to estimate the initial pose of the headset without knowing its original position in relation to the circuit board.

10) The initial pose of the headset is sent to the subfunction Scene management/Virtual Scene Update to update the position of the virtual camera in the scene description. The subfunction Scene management/Virtual Scene Update updates the state of activated objects that define a behaviour (here the current two pins to connect are activated, and their position is updated according to the associated animation).

11) By using the subfunction 3D Rendering/Video Rendering the rendering engine passes through the scene description to create an image of the model of the circuit board with the current pins to connect as seen from the viewpoint of the virtual camera.

12) The image provided from the 3D renderer can be adapted to the specific optical characteristics of the headset by the subfunction Rendering Adaption/Optical see-through. The resulting adaptation is displayed on the screen of the headset.

13) The technician starts its activity of connecting the first two pins on the circuit board. Each time an image is captured by the built-in camera or a movement is detected by the subfunction World Capture/Orientation and movement, an update of the pose of user viewpoint is computed using the subfunction World Analysis/AR Device Tracking that makes use of the pose of headset at the previous frame to estimate the pose at the current frame. This estimation leads to an updated pose that is sent to the subfunction Scene management/Virtual Scene Update. The processing steps 9, 10, 11 and 12 are repeated until the employee signals to the application that the wiring is done.

14) The technician has connected the first pair of pins and reported it using the vocal keyword "Done". This keyword is recognized and interpreted by the subfunction User Interactions/Vocal as the command "Done" which is sent to the subfunction Scene Management/Interaction Technique.

15) The subfunction Scene Management/Interaction Technique maps the command "Done" to the action "Next step" for which an event is sent to virtual scene update.

NOTE: What is interesting here is that the interaction can be easily swapped by another one by simply adding a new interaction subfunction such as a User Interactions/3D gesture which can recognized an "air tap gesture" command, and by adding the mapping from "air tap gesture" command to the "Next Step" action in the subfunction Scene Management/Interaction Technique**.**

16) When the subfunction Scene management/Virtual Scene Update receives an action event "Next Step", it will advance the scenario by one step by deactivating here the current pair of pins and activating the next one. Then, processing steps 9, 10, 11, 12 are repeated until the employee reports to the application that the wiring is done.

7.3 Manufacturing procedure

7.3.1 Use case description

This use case describes an AR system assisting an operator during a manufacturing procedure in order to improve the performance of the operators and reduce the manufacturing and assembly errors. This use case consists of the following steps:

- During preparation:
    - Preparing the 3D model of the workstations.
    - Preparing general information for each workstation.
    - Preparing content related to instructions for each workstation in three modes: textual, video and synthetic 3D animations (several animations can be defined according to various operator's experience levels).
    - Extracting recognition and localization features from the CAD models of the workstations.
    - Extracting localization features from CAD models of the tools used for the procedures.
    - Extracting the recognition features representing the complete procedure.
    - Placing instructions based on synthetic 3D animation in the reference of the workstation coordinate system.
- During the AR experience:
    - Workstation identification: allows to display the AR manufacturing procedure corresponding to the workstation facing the user.
    - Workstation presentation: General information concerning the workstation are presented in AR to the user.
    - Manufacturing procedure: Each step of the manufacturing procedure corresponding to the workstation are displayed in AR. If a dedicated tool is required for the step, the AR system highlight it (see Figure 7). Instructions are displayed to the operator thanks to a textual description, a video and a synthetic 3D animation perfectly positioned with the real workstation (see Figure 8). When the procedure is completed, the system checks if the result of the procedure corresponds to what was expected.

Figure 7: Current tool detection

Figure 8: Assembly task animation

7.3.2 Use case implementation

Figure 9 shows workflow diagram describing an example of implementation of the AR framework for the use case "Manufacturing procedure".

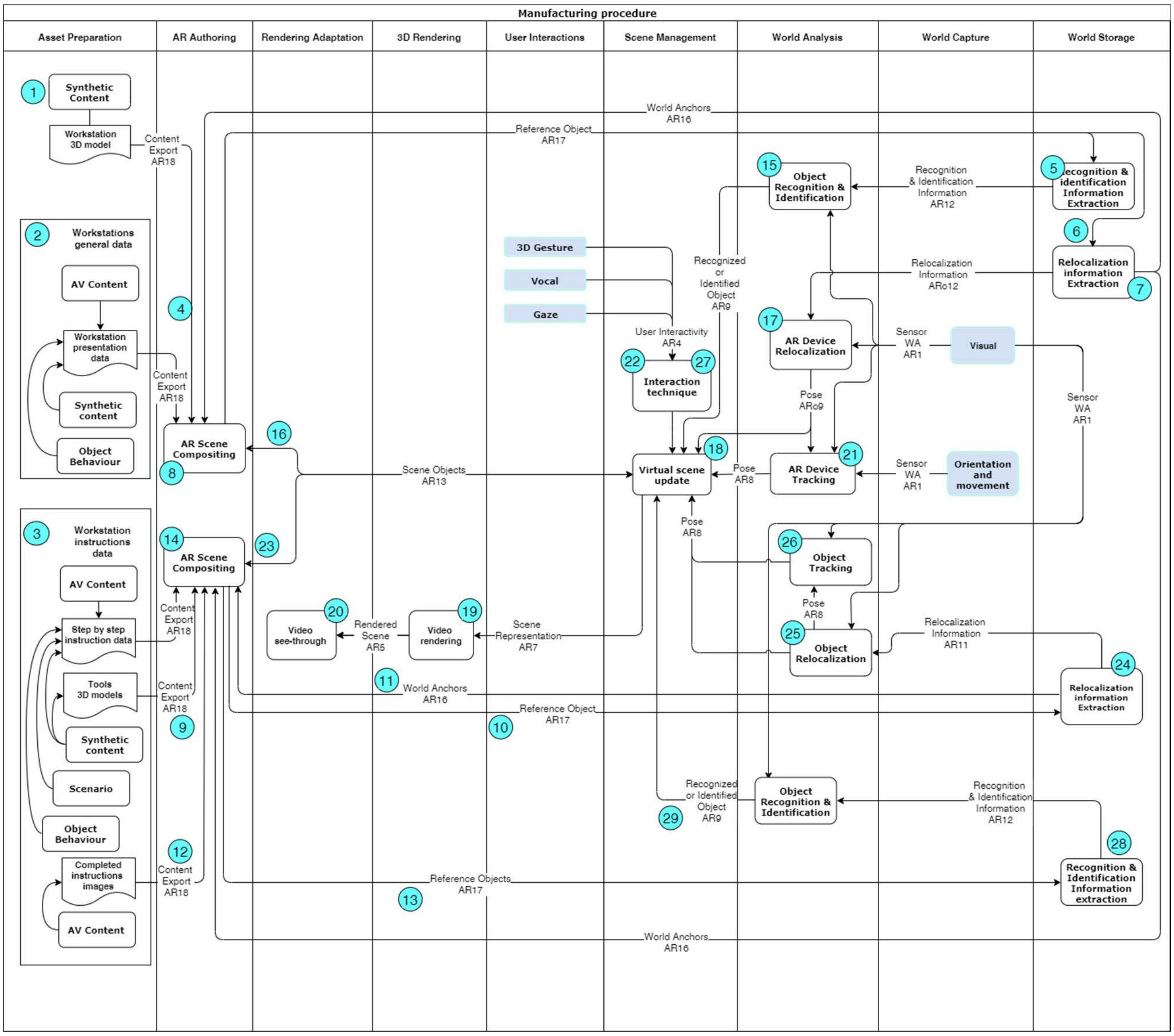

Figure 9: Workflow diagram of the subfunctions implemented for the use case "Manufacturing procedure"

Preparation stage (independent actions):

1) With a 3D modeling software a 3D graphic designer produces the 3D model of the workstation containing the supermarkets with boxes and the tools needed to complete all the tasks of the specific workstation (subfunction Asset Preparation/Synthetic Content). This 3D model is defined in a given unit (e.g. meter) and matches the size of the real workstation for registration of the digital media with the real world.

2) With the help of a dedicated authoring tool, the general workstation data is created by the workstation expert (subfunction Asset Preparation/Synthetic Content and Asset Preparation/AV Content). In addition to creating the digital media related to the workstation, the expert also defines the behaviour of the digital media (subfunction Asset Preparation/Objects behaviour) in order for the system to react consistently based on operator's experience, preferences and context (e.g. workstation or task number).

3) By using the same authoring tool, or a separate one (depending on the architecture of the system), the expert creates the step by step instructions for the workstation (subfunction Asset Preparation/Synthetic Content and Asset Preparation/AV Content). In addition, the expert authors how the digital media corresponding to these instructions is displayed to the user: when, where, for how long and how the operator can interact with them (subfunction Asset Preparation/Objects behaviour and Asset Preparation/Scenario). This step is needed for the system to correctly display the augmentation considering all the variables, including operator's vocal commands, gestures, gaze and movement.

4) The AR service developer exports the 3D Model of the workstation to be recognized and localized and adds it to the list of potential reference objects with the subfunction AR Authoring/ AR Scene Compositing.

5) The exported 3D model of the workstation is uploaded to a server to extract recognition information (subfunction World Storage/Recognition & Identification Information Extraction).

6) The exported 3D model of the workstation is uploaded to a server to extract relocalization information (subfunction World Storage/Relocalization Information Extraction).

7) An anchor corresponding to the workstation's coordinate reference system is sent back to the subfunction AR Authoring/AR Scene Compositing.

8) The AR service developer positions the different workstation instructions in the 3D space related to the anchor of the workstation.

9) The AR Service developer exports the 3D models of tools to be localized during the procedure and add them to the list of potential reference objects with the subfunction AR Authoring/AR Scene Compositing.

10) The exported 3D models of the tools are uploaded to a server to extract the relocalization information (subfunction World Storage/Recognition & Identification Information Extraction).

11) An anchor corresponding to each tool's coordinate reference system is sent back to the subfunctions AR Authoring/AR Scene Compositing.

12) The AR service developer exports the images corresponding to the expected result when the procedure is completed.

13) The exported images, corresponding to the expected result when the procedure is completed, are uploaded to a server to extract recognition information (subfunction World Storage/Recognition & Identification Information Extraction).

14) The AR service developer positions the AR scene relative to the anchor of the workstation and the anchors of different 3D models of tools.

AR experience (synchronous actions):

15) The operator starts the AR service on an AR device. The recognition information is used by the subfunction World Analysis/Object recognition & Identification in order to detect which workstation is currently in front of the operator. The camera captures the real world, and when the workstation is recognized,, data about this recognition (e.g. the object ID) is transmitted to the subfunction Scene Management/Virtual Scene Update.

16) Based on the recognition data, a request is sent to download the workstation presentation data scenario from the subfunction AR Authoring/AR Scene Compositing**.** The scene objects corresponding to this scenario are transmitted to the subfunction Scene Management/Virtual Scene Update which will load them in the current scene graph representing the elements to be rendered.

17) The AR device's camera captures images (subfunction World Capture/Visual) which are sent to the subfunction World Analysis/AR Device Relocalization. This subfunction uses the relocalization information provided by the subfunction World Storage/Relocalization Information Extraction without knowing its original position, to estimate the pose of the AR device.

18) The pose of the AR device is sent to the subfunction Scene management/Virtual Scene Update to update the position of the virtual camera in the scene description.

19) Using the subfunction 3D Rendering/Video Rendering the rendering engine renders the scene description to create an image of the workstation.

20) The image provided from the 3D renderer can be adapted by the subfunction Rendering Adaptation/Video see-through.

21) Each time an image is captured by the built-in camera with the subfunction World Capture/Visual or a movement is detected by the subfunction World Capture/Orientation and movement, an update of the pose of user's viewpoint is estimated using the subfunction World Analysis/AR Device Tracking that makes use of the pose of the AR device at the previous frame to estimate its pose at the current frame. This estimation leads to an updated pose that is sent to the subfunction Scene management/Virtual Scene Update. The processing steps 18, 19, 20 and 21 are repeated until the tracking fails. In case of a tracking failure, a new relocalization (step 17) is launched.

22) In order to initiate the step-by-step instructions, the technician uses a voice command, a 3D gesture and/or a gaze interaction to select a procedure in the graphic interface displayed in overlay. The technician's command is interpreted by the subfunction Scene Management/Interaction Technique as an action "start step-by-step procedure", which is sent to the subfunction Scene Management/Virtual Scene Update.

23) The subfunction Scene Management/Virtual Scene Update receiving the action "start step-by-step procedure" requests the corresponding AR content from the subfunction AR Authoring/AR Scene Compositing**.** The scene objects corresponding to the selected procedure are sent to the subfunction Scene Management/Virtual Scene Update. The system loads them in the current scene graph representing the elements to render. Note that steps 18, 19, 20 and 21 are being processed continuously.

24) The relocalization information of the tools used for the selected procedure can be sent by the World Storage/Relocalization Information Extraction to the subfunction World Analysis/Object Relocalization.

25) The subfunction **World Capture/Visual** transmits an image captured by the camera to the subfunction World Analysis/Object Relocalization which localizes the tools. When a tool is localized, the subfunction World Analysis/Object Relocalization sends the pose of the tool to the subfunction Scene Management/Virtual Scene Update which will register the 3D model of the tool.

26) Each time an image is captured, an update of the pose of objects is estimated using the subfunction World Analysis/Object Tracking that makes use of the pose of the objects at the previous frame to estimate their pose at the current frame. This estimation leads to an updated pose that is sent to the subfunction Scene management/Virtual Scene Update. The processing steps 18, 19, 20, 21 and 26 are repeated.

27) Upon completion of the step-by-step procedure, the operator reports it to the AR system using a voice command, a 3D gesture and/or a gaze interaction.

28) The recognition information of the completed procedure is sent by the World Storage/Recognition & identification Information Extraction to the subfunction World Analysis/Object Recognition & Identification.

29) The World Analysis/Object Recognition & Identification subfunction uses the recognition information previously provided to assess if the current state of the workstation corresponds to the expected results. If the expected state of the workstation is recognized, an event is sent to the subfunction Scene Management/Virtual Scene Update which updates the internal scene representation. According to the scenario, a feedback can be displayed to the user to inform him that the procedure has been successfully completed.

7.4 Collaborative design review

7.4.1 Use case description

This use case describes an AR collaborative design review of an industrial product where a first user equipped with an AR optical see-through headset starts to scan a room and puts a virtual 3D model of an industrial product on a real desk. A second user located in the same room also equipped with an AR optical see-through headset is able to visualize the virtual 3D model of the industrial product on the desk. Thus, both users can review the design of this product. Both users are co-located, meaning that when a user points at a part of the product, the other user knows exactly which part this user is pointing at. This use case consists of the following steps.

During preparation:

- Extracting the 3D model of the industrial product from a Product Lifecycle Management system.

During the AR experience:

- First user:
  - Scanning the room.
  - Placing the 3D model of the industrial product on a real desk.
  - Visualizing the 3D model of the industrial product.
- Second User:
  - Visualizing the 3D model of the industrial product

7.4.2 Use case implementation

Figure 10 shows the workflow diagram describing an example of the implementation of the AR framework for the use case "Collaborative design review."

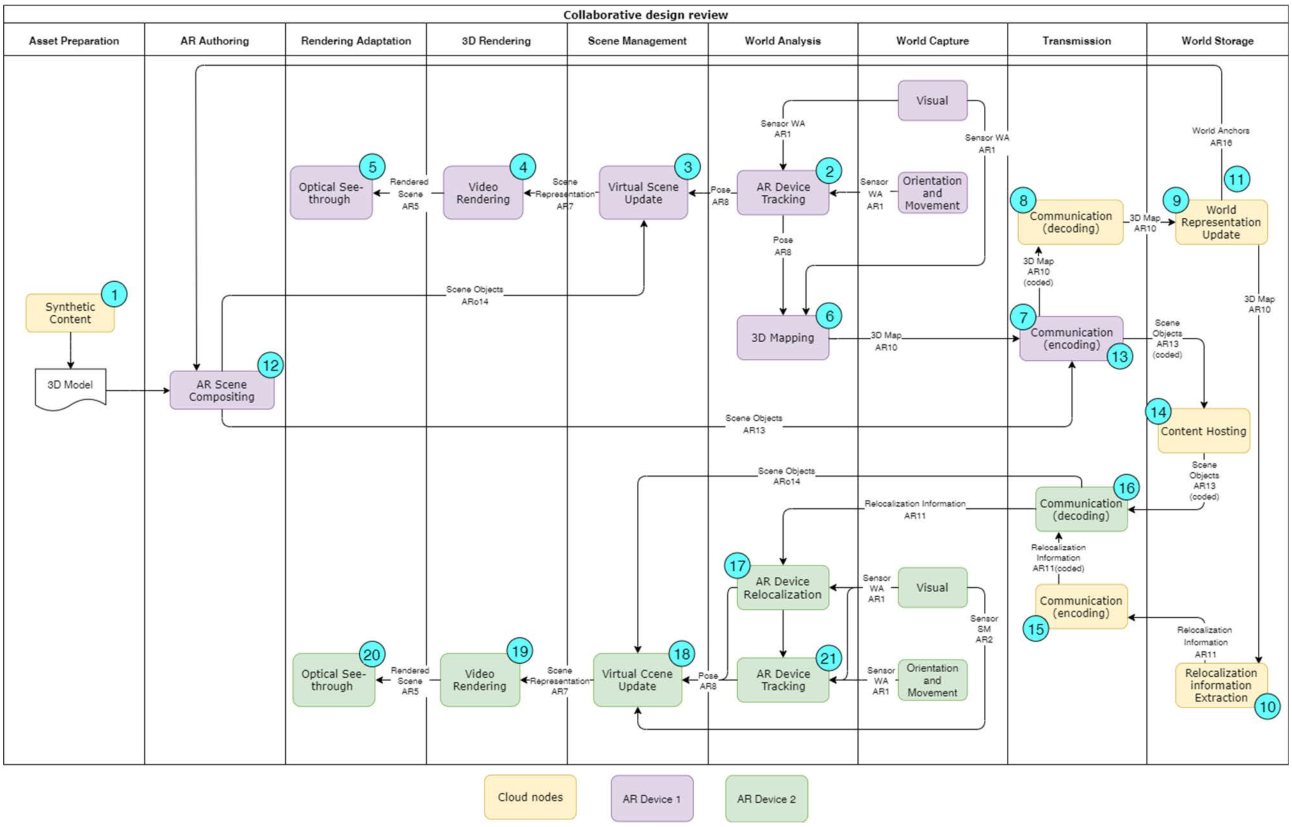

Figure 10: Workflow diagram of the subfunctions implemented for the use case "Collaborative design review"

Preparation phase:

1) The 3D model of the industrial product is exported from a Product Lifecycle Management system (subfunction Asset Preparation/Synthetic Content). This 3D model includes information about its 3D representation and also its scale.

AR experience:

2) The first user launches the application for AR collaborative design review. The built-in camera of the AR device starts capturing the room. Each time an image is captured by the subfunction **World Capture/Visual** or a movement is detected by the subfunction World Capture/Orientation and movement, the change of the user's viewpoint is estimated using the subfunction World Analysis/AR Device Tracking that makes use of the pose of the built-in camera in the previous frame to estimate the pose in the current frame (the pose of the AR device at the first frame establishing the reference coordinate system of the world).

3) The refreshed pose of the built-in camera of the AR device is sent to the subfunction Scene management/Virtual Scene Update to update the position of the virtual camera in the scene description.

4) By using the subfunction 3D Rendering/Video Rendering, the rendering engine renders the scene description to create an image of the AR scene as seen from the viewpoint of the virtual camera. For the moment, the AR scene is empty and no content is rendered.

5) The image provided by the 3D renderer can be adapted by the subfunction Rendering Adaption/Optical see-through to be consistently displayed on the screen(s) of the AR headset of the first user. Note that the steps 2, 3, 4 and 5 are running in a loop.

6) At regular intervals, a 3D reconstruction of the room is processed with the subfunction **World Analysis/3D Mapping.** To do this, keypoints are extracted from the image captured by the built-in camera, and descriptors are computed for each keypoint. Then, keypoints are matched between the current and the previous images based on the similarity of their descriptors. The pairs of keypoints that match are triangulated based on the pose of the previous and current viewpoint to build a 3D point cloud representing the room. A 2D descriptor is attached to each 3D point of the cloud. This descriptor is determined from the descriptors of the keypoints used to create the 3D point.

NOTE 1: At this point, there could be an extension of this use case. To inform the user about the parts of the room that are already scanned, a 3D mesh can be built with the subfunction **World Analysis/Scene Meshing** by using the point cloud produced by the subfunction **World analysis/3D Mapping.** This mesh can then be added to the AR scene using the subfunction **AR Authoring/AR Scene Compositing.**

7) Since the size of this point cloud can be relatively large, it is encoded to be transmitted on the cloud with the subfunction **Transmission/Communications.**

8) A cloud node decodes the point cloud with the subfunction **Transmission/Communications**.

9) This decoded point cloud is then conveyed to the subfunction **World Storage/World Representation**. Ideally the subfunction is on the same cloud node in order to avoid a new encoding and decoding process. The subfunction **World Storage/World Representation** handles a global and shared representation of the real world, here the room. For the initialization, since the representation of the room is empty, a simple copy of the received point cloud is applied. The steps 6, 7, 8 and 9 are repeated continuously to extend the map of the room.

10) Each time the world representation is updated, the subfunction **World Storage/Relocalization Information Extraction** is executed to provide other AR systems with the information required to relocalize the built-in camera of their AR device in the real world.

11) When the scan of the room is complete, world anchors in the coordinate reference system of the map of the room are sent to the subfunction **AR Authoring/AR Scene Compositing.**

12) The AR system of the first user exports the 3D model of the industrial product created in step 1 and positions it in the real world relatively to a world anchor with the subfunction **AR Authoring/AR Scene Compositing.**

NOTE 2: At this point, there could be an extension of this use case. An authoring graphic interface can be defined with the function **Assets Preparation** for adding a new 3D model to the AR scene. This can be achieved with specific behaviours attached to synthetic contents added to the AR scene with the subfunction **AR Authoring/AR Scene compositing.**

13) The composited scene with the 3D model of the industrial product positioned in the real space is encoded with the subfunction **Transmission/Communications** to be transmitted to a cloud node.

14) The cloud node running subfunction **AR Authoring/Content Hosting** will host the encoded scene. The encoded scene can be transmitted to and decoded by other AR devices to offer a shared AR experience.

15) The second AR user seeks to participate in the collaborative design review. A request is sent from the user's AR system to the subfunction **Transmission/Communication** to encode the relocalization information computed in step 10 and to transmit it to the AR device of the second user.

16) Another request is sent to the subfunction **AR Authoring/Content Hosting** to download the AR scene of the industrial product. The AR device decodes both the relocalization information and the AR scene of the industrial product with the subfunction **Transmission/Communication**.

17) The built-in camera of the AR device of the second user captures an image (subfunction World Capture/Visual) which is sent to the subfunction World Analysis/AR Device Relocalization. This subfunction uses the relocalization information provided by the subfunction Transmission/**Communication** to estimate the initial pose of built-in camera of the AR device in relation to the room without knowing its original position.

18) The initial pose of the built-in camera of the AR device of the second user is sent to the subfunction Scene management/Virtual Scene Update to update the position of its virtual camera in the scene description.

19) The subfunction 3D Rendering/Video Rendering renders the scene description to create an image of the model of the industrial product as seen from the viewpoint of the virtual camera.

20) The image provided by the 3D renderer is adapted by the subfunction Rendering Adaption/Optical see-through to be consistently displayed on the screen(s) of the AR system of the second user.

21) Each time an image is captured by the subfunction **World Capture/Visual** or a movement is detected by the subfunction World Capture/Orientation and movement, the change of the user's viewpoint is estimated using the subfunction World Analysis/AR Device Tracking that makes use of the pose of the built-in camera in the previous frame to estimate the pose in the current frame. This new pose estimation is sent to the subfunction Scene management/Virtual Scene Update. The processing steps 19, 20, and 21 are running in a loop until the second user complete the review.

7.5 Factory inspection based on an ARCloud

7.5.1 Use case description

This use case describes an inspection task using a visualization based on augmented reality where two inspectors carry out an inspection in the factory to check that the entire production tool is in working order. These inspectors visualize some data captured by smart sensors equipped at the different machines of the factory (e.g. pressure meters, temperature sensors or speedometers). The data of the smart sensors are displayed through a dynamic 3D graphic interface perfectly localized at the position of the real sensors. One inspector wears an optical see-through AR headset with high on-board computational resources while the second inspector wears light optical see-through AR glasses with very few computational resources. An edge cloud is available inside the factory to distribute processing requiring very low latency and a company cloud is accessible to support processing on confidential data that does not required low latency. Both AR devices worn by inspectors will also update the 3D map of the factory area during their inspection. This update of the 3D map will benefit to other workers who use AR systems. To simplify the implementation sample, it is assumed that a 3D map of the factory area already exists and that the synthetic content to display the data of the smart sensors have already been created and are well positioned in the real world thanks to anchors related to the map of the factory area (for more information on anchors, see implementation sample related to the collaborative design review in clause 7.4).

This use case consists of the following steps:

- During the inspection of the first inspector equipped with a high resources AR device:
    - Relocalizing the AR device (locally on the device) based on the 3D map of the factory area already captured.
    - Tracking the AR device and displaying the data (locally on the device) captured by the smart sensors through 3D graphic interfaces perfectly positioned on the different machines.
    - Mapping the factory area (locally on the device).
    - At regular intervals, transmitting the local map to update a global map of the factory area stored in a company cloud and shared by all AR users.

- During the inspection of the second inspector equipped with a low resources AR device:
  - Relocalizing the AR device (in an edge cloud) based on the 3D map of the factory area already captured.
  - Rendering (in an edge cloud) and displaying (locally on the device) the data captured by the smart sensors through 3D graphic interfaces perfectly positioned on the different machines.
  - Tracking the AR device (locally on the device).
  - Mapping the factory area (in an edge cloud node).
  - At regular intervals, transmitting the local map to update a global map of the factory area stored in a company cloud and shared by all AR users.

7.5.2 Use case implementation

Figure 11 shows the workflow diagram describing an example of the implementation of the AR framework for the use case "Factory inspection based on an ARCloud".

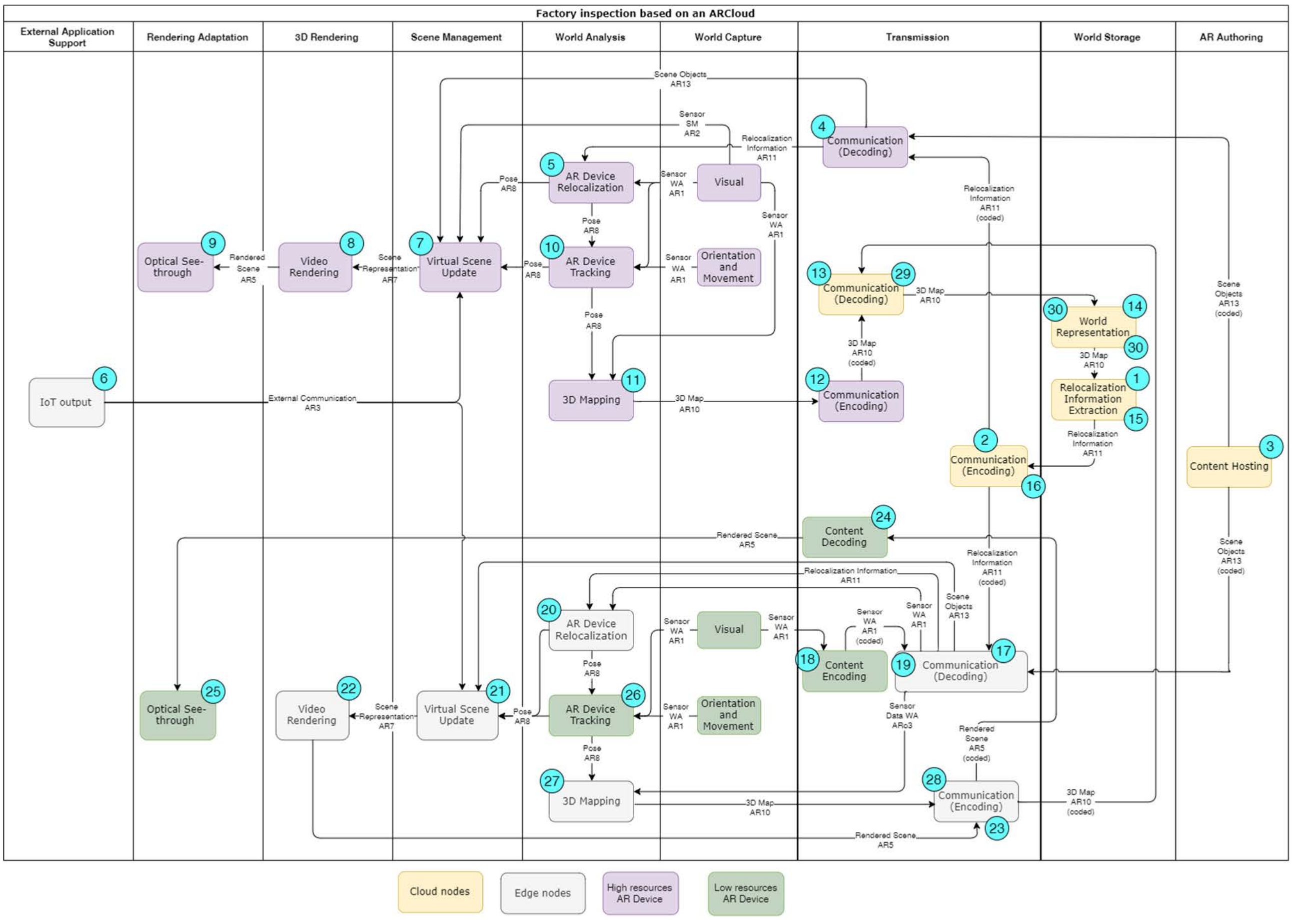

Figure 11: Workflow diagram of the subfunctions implemented for the use case "Factory inspection based on an ARCloud"

Inspection of the first inspector equipped with a high-resource AR device:

1) The first inspector starts the inspection by launching the AR service on his AR device. A request is sent from the inspector's AR device to the subfunction **World Storage/Relocalization Information Extraction** to extract the relocalization information supported by the AR system. This relocalization information is extracted from the 3D map of the factory area stored by the **World Storage/World Representation Storage** subfunction. Note that it is assumed in this use case that the 3D map of the factory area has already been built over time by other AR devices.

2) The relocalization information is encoded in the company cloud with the **Transmission/Communication** subfunction to be transmitted to the AR device of the first inspector.

3) The subfunction **AR Authoring/Content Hosting** stores an encoded version of the AR scene including 3D graphic user interfaces used to display the flow of data provided by the smart sensors of the factory (e.g. virtual dials showing a pressure, a speed, or a temperature). For instance, a 3D graphic user interface representing a pressuremeter consists of a 3D model of a dial on which a behaviour is attached to move a needle according to the sensor data transmitted by the **External Application Support** function.

4) The AR device of the first inspector decodes both the relocalization information and the AR scene with the subfunction **Transmission/Communication**.

5) The built-in camera of the AR device of the first inspector captures an image (subfunction World Capture/Visual) which is sent to the subfunction World Analysis/AR Device Relocalization. This subfunction uses the relocalization information extracted by the **World Storage/Relocalization Information Extraction** subfunction and decoded by the Transmission/Communication subfunction to estimate the initial pose of the built-in camera of the AR device in relation to the reference system of the factory area without knowing its original position.

6) The data captured by the smart sensors equipped at the machines of the factory are managed in real time by an external application and adapted to the AR experience with the function **External Application Support**. This function transmits the adapted data to the subfunction **Scene Management/Virtual Scene Update.**

7) The initial pose of the built-in camera of the AR device of the first inspector user is sent to the subfunction Scene management/Virtual Scene Update to update the position of the virtual camera in the AR scene description. Also, the subfunction Scene management/Virtual Scene Update updates the AR scene description by injecting the adapted data captured by the smart sensors into their respective 3D graphic user interfaces. For instance, the data corresponding to a temperature of a sensor will update the position of the needle of a virtual temperature dial. This update is managed by the behaviour attached to the synthetic content representing the temperature dial.

8) The subfunction 3D Rendering/Video Rendering renders the scene description to create an image of the 3D graphic user interfaces of the meters as seen from the viewpoint of the virtual camera.

9) The image provided by the 3D renderer is adapted by the subfunction Rendering Adaption/Optical see-through to be consistently displayed on the screen(s) of the AR device of the first inspector.

10) Each time an image is captured by the subfunction **World Capture/Visual** or a movement is detected by the subfunction World Capture/Orientation and Movement, the change of the user's viewpoint is estimated using the subfunction World Analysis/AR Device Tracking that makes use of the pose of the built-in camera in the previous frame to estimate the pose in the current frame. This new pose estimation is sent to the subfunction Scene management/Virtual Scene Update. The processing steps 7, 8, 9, and 10 are running in a loop until the first inspector completes the inspection.

11) At regular intervals, a 3D reconstruction of the part of the factory area visited by the first inspector is processed with the subfunction **World Analysis/3D Mapping.** To do this, keypoints are extracted from the image captured by the built-in camera, and descriptors are computed for each keypoint. Then, keypoints are matched between the current and the previous images based on the similarity of their descriptors. The pairs of keypoints that match are triangulated based on the pose of the previous and current viewpoint to build a 3D point cloud representing the factory area. A 2D descriptor is attached to each 3D point of the cloud. This descriptor is determined from the descriptors of the keypoints used to create the 3D point.

12) Since the size of this point cloud can be relatively large, it is encoded to be transmitted on the company cloud with the subfunction **Transmission/ Communications**.

13) A cloud node decodes the 3D point cloud with the subfunction **Transmission/Communications**.

14) This decoded 3D point cloud is then conveyed to the subfunction **World Storage/World Representation**. Ideally the subfunction is on the same cloud node in order to avoid a new encoding and decoding process. The subfunction **World Storage/World Representation** handles a global and shared representation of the real world, here the factory area. The partial 3D map reconstructed by the AR device of the first inspector is merged with the whole map already handled by the subfunction **World Storage/World Representation** to keep it up-to-date. Thus, if objects have moved in the factory area, the map is updated accordingly. The steps 11, 12, 13, and 14 are repeated continuously to update the map of the factory area.

15) Each time the world representation is updated, the subfunction **World Storage/Relocalization Information Extraction** is executed to provide other AR systems with the information required to relocalize the built-in camera of their AR device in the real world.

Inspection of the second inspector equipped with a low-resource AR device:

16) The second inspector equipped with a low-resource AR device seeks to participate in the inspection. Both the relocalization information and the AR scene composed of 3D graphic user interfaces used to show the flow of data streamed by the smart sensors are encoded in the cloud with the subfunction **Transmission/Communication.** The encoded relocalization information and AR scene are transmitted to an edge node which will handle the processing related to the AR device of the second inspector.

17) An edge node decodes both the relocalization information and the AR scene with the subfunction **Transmission/Communication**.

18) The built-in camera of the AR device of the second inspector captures an image (subfunction World Capture/Visual) which is encoded and transmitted to an edge node with the subfunction Transmission/**Communication**.

19) The edge node receiving the encoded image captured by the built-in camera of the AR device of the second inspector decodes the image with the subfunction **Transmission/Communication**.

20) The subfunction **World Analysis/AR Device Relocalization** running on the edge node uses the relocalization information and the captured image provided by the subfunction Transmission/**Communication** to estimate the initial pose of the built-in camera of the second inspector's AR device in relation to the reference system of the factory area without knowing its original position.

21) The initial pose of the built-in camera of the AR device of the second inspector is sent to the subfunction Scene management/Virtual Scene Update to update the position of the virtual camera in the AR scene description. Also, the subfunction Scene management/Virtual Scene Update updates the AR scene description by injecting the adapted data captured by the smart sensors into their respective 3D graphic user interfaces. For instance, the data corresponding to a temperature of a sensor will update the position of the needle of a virtual temperature dial. This update is managed by the behaviour attached to the synthetic content representing the temperature dial.

22) The subfunction 3D Rendering/Video Rendering renders on the edge node the scene description to create an image of the 3D graphic user interfaces of the meters as seen from the viewpoint of the virtual camera.

23) The rendered image is sent to the subfunction **Transmission/Communication** running on an edge node to transmit it to the AR device.

24) The subfunction **Transmission/Communication** running on the AR device of the second inspector decodes the image to send it to the subfunction **Rendering Adaptation/Optical See-Through.**

25) The image provided by the subfunction **Transmission/Communication** is adapted by the subfunction Rendering Adaption/Optical see-through to be consistently displayed on the screen(s) of the AR device of the second inspector.

26) Each time an image is captured by the subfunction **World Capture/Visual** or a movement is detected by the subfunction World Capture/Orientation and movement, the change of the user's viewpoint is estimated using the subfunction World Analysis/AR Device Tracking that makes use of the pose of the built-in camera in the previous frame to estimate the pose in the current frame. This new pose estimation is sent to the subfunction Scene management/Virtual Scene Update. The processing steps 26, 21, 22, 23, 24 and 25 are running in a loop until the second inspector completes the inspection.

27) At regular intervals, a 3D reconstruction of the part of the factory area visited by the second inspector is processed with the subfunction **World Analysis/3D Mapping.** To do this, keypoints are extracted from the image captured by the built-in camera and streamed on the edge node, and descriptors are computed for each keypoint. Then, keypoints are matched between the current and the previous images based on the similarity of their descriptors. The pairs of keypoints that match are triangulated based on the pose of the previous and current viewpoint to build a 3D point cloud representing the part of the factory area visited by the second inspector. A 2D descriptor is attached to each 3D point of the cloud. This descriptor is determined from the descriptors of the keypoints used to create the 3D point.

28) Since the size of this 3D point cloud can be relatively large, it is encoded to be transmitted on the company cloud with the subfunction **Transmission/ Communications**.

29) A node of the company cloud decodes the 3D point cloud with the subfunction **Transmission/Communication**.

30) This decoded 3D point cloud is then conveyed to the subfunction **World Storage/World Representation**. Ideally the subfunction is on the same cloud node in order to avoid a new encoding and decoding process. The subfunction **World Storage/World Representation** handles a global and shared representation of the real world, here the factory area. The partial 3D map reconstructed by the AR device of the second inspector is merged with the whole map already handled by the subfunction **World Storage/World Representation** to keep it up-to-date. Thus, if objects have moved in the factory area, the map is updated accordingly. The steps 27, 28, 29, and 30 are repeated continuously to update the map of the factory area.

7.6 Usability Evaluation of Virtual Prototypes

7.6.1 Use case description

Technical devices, such as coffee machines, printers, or ATMs, require a high usability. This ensures effectiveness, efficiency and satisfaction at the side of the end users, prevents errors, and serves as a baseline for good user experience. To achieve this, manufacturers should perform usability tests. This means, they should ask potential users of their devices to interact with early development versions of the device. By observing the users during the interaction, the manufacturers can derive the users' main issues and necessary improvements for the device under development.

The basis for such tests are prototypes of the technical devices. The construction of real world prototypes is usually rather expensive. Therefore, manufacturers consider performing user tests with virtual prototypes, e.g. in AR. This allows that users participate in such tests not (only) at the manufacturer's premises, but at home. In addition, the approach allows performing unsupervised tests, at any time, and with a large number of users what provides more reliable results. This type of user test is usually called remote usability test.

To be able to analyse the results of remote usability tests, users may be asked to fill in questionnaires. In addition, the usage of the virtual prototype is recorded, e.g. using video and audio recordings of the user and the AR scene during the usage. As this threatens the users' privacy, the recordings are often done anonymously on the level of AR internal events. This means, the AR application only records the actions the users take with the virtual prototype as well as other AR internal events, e.g. AR device movements, being relevant for the test. Independent of the type of the acquired data they are stored and analysed subsequently using a dedicated analysis tool. For questionnaires, these may be statistics programs. For video and audio recordings, there are dedicated tools to view the recordings and to mark, to comment, and to analyse interesting sequences. For other types of recordings, also corresponding tools for analysis exist. These can visualize statistical usage data, provide means of data analytics, or generate detailed reports on the usage and the usability of the virtual prototype.

To support a remote usability test of a virtual prototype the manufacturer first creates a CAD model of the device under development including interaction elements (e.g. buttons and knobs) as well as visualization elements (e.g. lights and displays). Then the manufacturer defines the intended functionalities of the device using, e.g. state machines. The states of the state machines are the states the device can take. They usually comprise a set of values visualized by the visualization elements. The transitions define how state changes can occur. State changes can occur through user interaction with one of the interaction elements or by other means, e.g. the expiry of a timeout.

When a manufacturer completes these specifications, it compiles a virtual prototype and provides it to users using mobile AR on their smartphone. The users display the prototype at home. They may also position the prototype at the intended location, e.g. they may position a virtual coffee machine in their kitchen. Then the users will interact with the prototype to try out certain interaction tasks. The interaction of the users is recorded as described above. Finally, the users may fill in a questionnaire. After the tests, the manufacturer analyses the data using the above mentioned tools and derives improvements for the prototype and the device under development. Then the manufacturer updates the virtual prototype, i.e. the CAD model and other specifications to start a new evaluation round.

This use case consists of the following steps:

- Test preparation:
  - The manufacturer creates the CAD model of the technical device under development.

- The manufacturer specifies the behaviour of individual elements of the technical device.
- The manufacturer defines the state machines representing the intended device functionalities including the states and transitions.
- The manufacturer defines the tasks that users will perform during a remote test.
- The manufacturer specifies the questions to be asked at the end of a test session.
- Finally, the manufacturer compiles a mobile AR application that contains the virtual prototype, the task specification, and the questionnaire. This application is to be run by the test users.

- Test execution using mobile AR:
  - The user opens the mobile AR application on her/his smartphone.
  - The user positions the virtual prototype at the desired location.
  - The application asks the user to perform the predefined tasks.
  - The user interacts with the virtual prototype according to the tasks while her/his interactions are recorded.
  - The user answers the final questionnaire and closes the application.
- Test analysis:
  - The manufacturer loads the recorded data into a corresponding analysis tool.
  - The manufacturer analyses the test and derives improvements for the virtual prototype.
  - The manufacturer adapts all specifications done in the test preparation to improve the virtual prototype and starts a new test execution.

7.6.2 Use case implementation

Figure 12 shows workflow diagram describing the implementation of both parts of the use case using the defined AR reference framework.

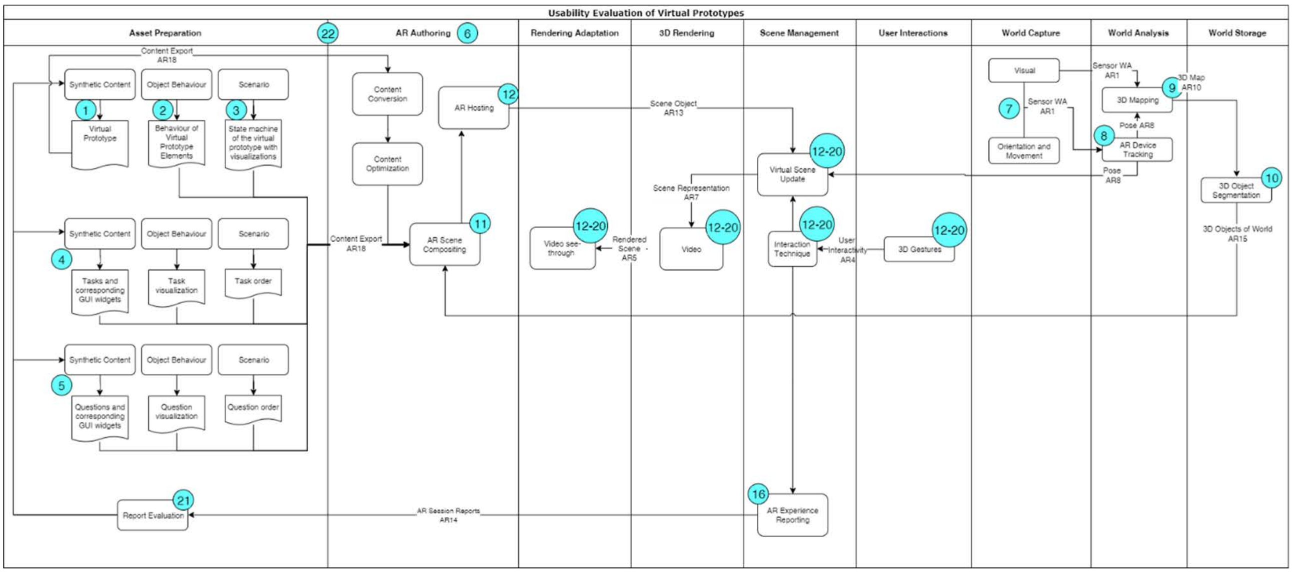

Figure 12: Workflow diagram of the subfunctions implemented for the use case "Usability Evaluation of Virtual Prototypes"

Test preparation:

1) The model of the virtual prototype results from the **Asset preparation** function. The manufacturer creates the model using CAD tools or similar software with the subfunction **Asset Preparation/Synthetic Content**. The virtual prototype is created in a form that allows a separation and naming of its individual components such as individual buttons, knobs, sliders, lights, and displays.

2) The manufacturer then specifies component specific behaviours for components of the virtual prototype using the **Asset Preparation/Object Behaviour** subfunction. This covers for example the possible rotation of knobs, the visualization of lights, or the contents of a display of the virtual prototype.

3) The manufacturer then defines the overall functionalities of the virtual prototype. This encompasses defining, e.g. states and transitions of a state machine. Every state herewith represents a combination of visualizations displayed by the virtual prototype and positions of knobs, buttons, sliders, and other interaction elements. These definitions take place in the **Asset Preparation/Scenario** subfunction.

4) Next, the manufacturer defines the tasks that the users will perform with the virtual prototype. This covers the definition of the tasks themselves as well as GUI widgets for their presentation in AR. Both are done using the **Asset Preparation/Synthetic Content** subfunction. Potential animations for displaying and discarding the task descriptions are defined using the **Asset Preparation/Object Behaviour** subfunction. In addition, the manufacturer uses this subfunction to specify interaction techniques for skipping tasks or switching between them. The order of the tasks is defined using the **Asset Preparation/Scenario** subfunction.

5) Furthermore, the manufacturer defines the questions that the users will answer after having interacted with the virtual prototype. This covers the definition of the question text as well as GUI widgets for their presentation and the input of corresponding answers in AR. Both are done using the **Asset Preparation/Synthetic Content** subfunction. Potential animations for displaying and discarding the questions are defined using the **Asset Preparation/Object Behaviour** subfunction. In addition, the manufacturer uses this subfunction to specify interaction techniques for entering answers into the respective widgets and to switch between questions. The order of display for the questions and corresponding GUI widgets are defined using the **Asset Preparation/AR Scenario** subfunction.

6) Finally, the manufacturer defines the overall AR scene by combining all the specifications using the **AR Authoring**/**AR Scene Compositing** subfunction. This may incorporate using the **AR Authoring**/**Content Conversion** and **AR Authoring**/**Content Optimization** subfunction to transform the specifications created with external tools into a format that can be displayed and processed by the AR System. In addition, the manufacturer will reuse other synthetic contents, object behaviours, and AR scenarios that are reused from former AR scenes with a similar setup. This covers, e.g. GUI widgets, behaviours, and AR scenarios to position the virtual prototype relative to the real world. The fully defined AR scene is then stored by the **AR Authoring/Content Hosting** subfunction.

Test execution using mobile AR:

1) At startup, the built-in camera of the AR device is launched and the **World Capture** starts to provide sensor data from the real world to the **World Analysis** function.

2) Each time an image is captured by the subfunction **World Capture/Visual** or a movement is detected by the subfunction World Capture/Orientation and Movement, the change of the user's viewpoint is estimated using the subfunction World Analysis/AR Device Tracking that makes use of the pose of the built-in camera in the previous frame to estimate the pose in the current frame. This new pose estimation is sent to the subfunction Scene management/Virtual Scene Update to update the position of the virtual camera in the AR scene description. The processing steps 7 and 8 are running in a loop until the user completes the test.

3) A 3D reconstruction of the user environment is processed with the subfunction **World Analysis/3D Mapping.** To do this, keypoints are extracted from the image captured by the built-in camera, and descriptors are computed for each keypoint. Then, keypoints are matched between the current and the previous images based on the similarity of their descriptors. The pairs of keypoints that match are triangulated based on the pose of the previous and current viewpoint to build a 3D point cloud representing the user's environment. A 2D descriptor is attached to each 3D point of the cloud. This descriptor is determined from the descriptors of the keypoints used to create the 3D point.

4) The **Object Segmentation** subfunction detects planar surfaces in the real world based on the 3D map built at the previous step. It forwards the representations of the surfaces to the **AR Authoring** function.

5) The surface representations are added to the AR Scene with the **AR Authoring/AR Scene Compositing** subfunction. These new updates on the AR Scene are stored with the **AR Authoring/AR Hosting** subfunction.

6) The **Scene Management** updates its internal representation of the AR Scene according to the new updates including the detected surface representations hosted by the **AR Authoring/AR Hosting** subfunction. These surface representations will not be displayed, but will be used as colliders to ease the positioning of the virtual prototype on top of them. A dedicated GUI will help the user for positioning the virtual prototype. This GUI is retrieved from the **AR Authoring/Content Hosting** subfunction, added to the AR scene with the **Scene Management/Virtual Scene Update**, rendered by the **3D Rendering** function and displayed to the user with the **Rendering Adaptation** function.

7) After the user has selected the position of the virtual prototype (determined by the **Scene Management/Interaction Technique** subfunction using information from the **User Interactions** function) the **Scene Management** retrieves the virtual prototype from the **AR Authoring/Content Hosting** subfunction. The **Scene Management/Virtual Scene Update** subfunction adds the virtual prototype to the AR scene according to the user's specification and the **3D Rendering** and the **Rendering Adaptation** functions display it to the user.

8) The first task as defined at step 4 and hosted at step 6 is presented to the user by adding the corresponding objects to the AR scene with the **Scene Management/Virtual Scene Update** subfunction and by displaying them using the **3D Rendering** and the **Rendering Adaptation** functions.

9) The **Scene Management/Interaction Techniques** subfunction determines user actions with the virtual prototype using data captured by the **User Interactions** function.

10) At this stage, every interaction of the user is recorded using the **Scene Management/AR Experience Reporting** subfunction. This means, it stores every relevant event registered by the **Scene Management/Interaction Technique** subfunction as well as additional information such as pose updates of the AR device retrieved from the **World Analysis** function. The information is transmitted to the **Asset Preparation/Report Evaluation** subfunction via the **AR Session Reports** reference point.

11) Based on the user actions with the virtual prototype, the **Scene Management/Virtual Scene Update** subfunction determines the next state of the virtual prototype to be loaded based on the AR scene composed in step 6. Using this information, this subfunction changes the state of the prototype and enables or disables signaling lights of the prototype, changes textures of visualization elements, or plays animations. The result is rendered by the **3D Rendering** function and displayed by the **Rendering Adaptation** function. The steps 15, 16, and 17 are continuously repeated until the user completes the current task.

12) Then the next task is displayed at step 14 and the steps 15, 16, and 17 are repeated continuously until the user completes the virtual prototype evaluation.

13) After all tasks are completed, the **Scene Management/Virtual Scene Update** subfunction removes the virtual prototype and the task from the AR scene and adds the first question of the questionnaire, which is displayed by the **3D Rendering** and **Rendering Adaptation** functions. The **Scene Management/Interaction Technique** subfunction recognizes user actions for answering the questions using data captured by the **User Interactions** function. It sends the answers of the user to the **Asset Preparation/Report Evaluation** subfunction via the **AR Session Reports** reference point.

14) Finally the user closes the application.

Test analysis

1) The manufacturer uses the **Asset Preparation/Report Evaluation** subfunction to analyse the recordings of the user sessions. This may cover creating user action heat maps, identifying wrong actions that users took for a certain task, analysing incomplete task executions including break up points, as well as determining typical answers to certain questions in the questionnaire.

2) Based on this analysis, the manufacturer derives improvements for the virtual prototype and implements them using the **Asset Preparation** function and the **AR Authoring/AR Scene Compositing** subfunction.

3) Then the manufacturer starts a new evaluation cycle by compiling the adapted AR scene and providing it to users.

History

Document history		
V1.1.1	March 2020	Publication