

Moving Picture, Audio and Data Coding by Artificial Intelligence <u>www.mpai</u>.community

MPAI Technical Specification

Portable Avatar Format (MPAI-PAF)

V1.4

WARNING

Use of the technologies described in this Technical Specification may infringe patents, copyrights or intellectual property rights of MPAI Members or non-members.

MPAI and its Members accept no responsibility whatsoever for damages or liability, direct or consequential, which may result from the use of this Technical Specification.

Readers are invited to review <u>Notices and Disclaimers</u>.

© Copyright MPAI 2021-2025. All rights reserved.

Technical Specification Portable Avatar Format (MPAI-PAF) V1.4

1	Fore	Foreword2			
2	Introduction				
3	Scope				
4	Defi	nitions	6		
5	Refe	prences	7		
	5.1	Normative References	7		
	5.2	Informative References	7		
6	AI V	Vorkflows	8		
	6.1	Videoconference Client Transmitter	11		
7	AI N	Iodules	15		
	7.1	Audio-Visual Scene Creation	16		
	7.2	Audio-Visual Scene Rendering	17		
	7.3	Entity Body Description	19		
	7.4	Entity Face Description	20		
	7.5	Face Identity Recognition	22		
	7.6	Face Personal Status Extraction	24		
	7.7	Gesture Personal Status Extraction	26		
	7.8	Portable Avatar Demultiplexing	28		
	7.9	Portable Avatar Multiplexing	30		
	7.10	PS-Face Interpretation	33		
	7.11	PS-Gesture Interpretation	34		
	7.12	Personal Status Display	35		
	7.13	Service Participant Authentication	38		
	7.14	Visual Scene Creation	39		
8	Data	ı Types	41		
	8.1	Avatar	41		
	8.2	Body Descriptors	42		
	8.3	Face Descriptors	43		
	8.4	Gesture Descriptors	45		
	8.5	MoCap Object	46		
	8.6	Portable Avatar	46		

1 Foreword

The international, unaffiliated, non-profit *Moving Picture, Audio, and Data Coding by Artificial Intelligence (MPAI)* organisation was established in September 2020 in the context of:

- 1. **Increasing** use of Artificial Intelligence (AI) technologies applied to a broad range of domains affecting millions of people
- 2. Marginal reliance on standards in the development of those AI applications
- 3. **Unprecedented** impact exerted by standards on the digital media industry affecting billions of people

believing that AI-based data coding standards will have a similar positive impact on the Information and Communication Technology industry.

The design principles of the MPAI organisation as established by the MPAI Statutes are the development of AI-based Data Coding standards in pursuit of the following policies:

1. Publish upfront clear Intellectual Property Rights licensing frameworks.

- 2. <u>Adhere to a rigorous standard development process.</u>
- 3. <u>Be friendly</u> to the AI context but, to the extent possible, remain agnostic to the technology thus allowing developers freedom in the selection of the more appropriate AI or Data Processing technologies for their needs.
- 4. <u>Be attractive</u> to different industries, end users, and regulators.
- 5. <u>Address</u> five standardisation areas:
 - 1. *Data Type*, a particular type of Data, e.g., Audio, Visual, Object, Scenes, and Descriptors with as clear semantics as possible.
 - 2. *Qualifier*, specialised Metadata conveying information on Sub-Types, Formats, and Attributes of a Data Type.
 - 3. *AI Module* (AIM), processing elements with identified functions and input/output Data Types.
 - 4. *AI Workflow* (AIW), MPAI-specified configurations of AIMs with identified functions and input/output Data Types.
 - 5. *AI Framework* (AIF), an environment enabling dynamic configuration, initialisation, execution, and control of AIWs.
- 6. <u>Provide</u> appropriate Governance of the ecosystem created by MPAI Technical Specifications enabling users to:
 - 1. *Operate* Reference Software Implementations of MPAI Technical Specifications provided together with Reference Software Specifications
 - 2. *Test* the conformance of an implementation with a Technical Specification using the Conformance Testing Specification.
 - 3. *Assess* the performance of an implementation of a Technical Specification using the Performance Assessment Specification.
 - 4. *Obtain* conforming implementations possibly with a performance assessment report from a trusted source through the MPAI Store.

MPAI operates on four solid pillars:

- 1. The <u>MPAI Patent Policy</u> specifies the MPAI standard development process and the Framework Licence development guidelines.
- <u>Technical Specification: Artificial Intelligence Framework (MPAI-AIF) V2.1</u> specifies an environment enabling initialisation, dynamic configuration, and control of AIWs in the standard AI Framework environment depicted in Figure 1. An AI Framework can execute AI applications called AI Workflows (AIW) typically including interconnected AI Modules (AIM). MPAI-AIF supports small- and large-scale high-performance components and promotes solutions with improved explainability.



Figure 1 – The AI Framework (MPAI-AIF) V2 Reference Model

- 3. <u>Technical Specification: Data Types, Formats, and Attributes (MPAI-TFA) V1.2</u> specifies Qualifiers, a type of metadata supporting the operation of AIMs receiving data from other AIMs or from input data. Qualifiers convey information on Sub-Types (e.g., the type of colour), Formats (e.g., the type of compression and transport), and Attributes (e.g., semantic information in the Content). Although Qualifiers are human-readable, they are only intended to be used by AIMs. Therefore, Text, Speech, Audio, Visual, and other Data received by or exchanged between AIWs and AIMs should be interpreted as being composed of Content (Text, Speech, Audio, and Visual as appropriate) and associated Qualifiers. For instance, a Text Object is composed of Text Data and Text Qualifier. The specification of most MPAI Data Types reflects this point.
- 4. <u>*Technical Specification: Governance of the MPAI Ecosystem (MPAI-GME) V1.1</u> defines the following elements:</u>*
 - 1. <u>Standards</u>, i.e., the ensemble of Technical Specifications, Reference Software, Conformance Testing, and Performance Assessment.
 - 2. <u>Developers</u> of MPAI-specified AIMs and <u>Integrators</u> of MPAI-specified AIWS (Implementers).
 - 3. <u>MPAI Store</u> in charge of making AIMs and AIWs submitted by Implementers available to Integrators and End Users.
 - 4. <u>Performance Assessors</u>, independent entities assessing the performance of implementations in terms of Reliability, Replicability, Robustness, and Fairness.
 - 5. End Users.

The interaction between and among actors of the MPAI Ecosystem are depicted in Figure 2.



Figure 2 – The MPAI Ecosystem

2 Introduction

(Informative)

There is a long history of computer-created objects called "digital humans", i.e., digital objects having a human appearance when rendered. In most cases the underlying assumption of these objects has been that creation, animation, and rendering is done in a closed environment. Such digital humans had little or no need for standards.

In a communication and more so in a metaverse context, there are many cases where a digital human is not constrained within a closed environment thus requiring forms of standardisation. *Technical Specification: Portable Avatar Format* (MPAI-PAF) V1.4 – in the following also called MPAI-PAF V1.4 or MPAI-PAF – is a response to the requirements of new usage contexts.

MPAI-PAF specifies a standard for Portable Avatar Format (PAF) enabling a receiving party to render a digital human as intended by the sending party.

MPAI-PAF V1.4 specifies the Avatar-Based Videoconference (PAF-ABV) AI Workflow where: 1. **Client Transmitters** send PAFs containing:

- At the beginning: Avatar Models, Language Selector, and Speech Object and Face Object for participant authentication.
- Continuously: Avatar Descriptors, and Speech Objects to a Server.

2. Avatar Videoconference Server:

- At the beginning:
 - Selects an Environment, i.e., a meeting room and equips it with objects, i.e., meeting table and chairs.
 - Places Avatar Models around the table.
 - Distributes for each participant a PAF containing Environment, Avatar Models, and their positions to all receiving clients.
- Continuously sends to receiving clients:
 - Translated Speech from participants according to Language Selectors.
 - Sends PAFs containing Avatar Descriptors and translated Speech.

3. Client Receivers:

- At the beginning: receive Environment and PAFs containing Avatar Models and Language Selectors from the server.
- Continuously from the server:
 - Receive PAFs containing Avatar Descriptors and translated Speech.
 - Create Audio and Visual Scene Descriptors.
 - Render the Audio-Visual Scene as seen from the human participant-selected Point of View.

In all Chapters and Sections, Terms beginning with a capital letter are defined in <u>*Table 1*</u> if they are specific to this Technical Specification. All MPAI-defined Terms are accessible <u>online</u>. All Chapters, and Sections are Normative unless they are labelled as Informative.

3 Scope

Technical Specification: Portable Avatar Format (MPAI-PAF) V1.4 specifies:

- 1. The *Portable Avatar Format* and related *Data Types* enabling a receiver to decode and render an Avatar and its Environment as intended by the sender.
- 2. The *Personal Status Display* Composite AI Module having the function of converting Text, Personal Status, and Avatar Model to a Portable Avatar.
- 3. The *Audio-Visual Scene Rendering* Composite AI Module having the function of rendering Audio-Visual Scene Descriptors and/or Portable Avatars as Audio and Visual information.
- 4. The AI Framework (MPAI-AIF)-conforming *AI Workflows and AI Modules* of the Avatar-Based Videoconference Use Case also using Data Types and AI Modules from other MPAI Technical Specifications.
- The Use Case normatively defines:
- 1. The <u>Functions</u> of the AIWs and of the AIMs.
- 2. The <u>Connections</u> between and among the AIMs
- 3. The <u>Semantics</u> and the <u>JSON Schemas</u> of the input and output data of the AIW and the AIMs.

The word *normatively* implies that an Implementation claiming Conformance to:

- 1. An AIW, shall:
 - 1. Perform the function specified in the relevant Section of <u>AI Workflows</u>.
 - 2. Use AIMs connected with the topology and connections conforming with the <u>AI</u> <u>Workflows</u> specifications.

- 3. Have input and output data with the formats specified by the relevant <u>AI Workflows</u> specifications.
- 2. An AIM, shall:
 - 1. Perform the AIM function specified in the relevant Section of <u>AI Modules</u> and AI Modules of MPAI-PAF V1.4 and of other Technical Specifications.
 - 2. Be composed of AIMs specified in the relevant Section of <u>AI Modules</u> and AI Modules of MPAI-PAF V1.4 and of other Technical Specifications in case the AIM is Composite.
 - 3. Receive and produce the data specified in the relevant Section of <u>Data Types</u> of MPAI-PAF V1.4 and of other Technical Specifications

Users of this Technical Specification should note that:

- 1. Implementers may use the Reference Software part of this Technical Specification to develop their Implementations.
- 2. The Conformance Testing specification can be used to test the conformity of an Implementation to this Standard.
- 3. Performance Assessors can assess the level of Performance of an Implementation based on the Performance Assessment associated with the MPAI-PAF Technical Specification.
- 4. Implementers and Users should consider the notices and disclaimers.

This Technical Specification includes the following elements:

- 1. Scope (This Chapter)
- 2. Definitions
- 3. References
- 4. AI Workflows for Avatar-Base Videoconference
- 5. AI Modules
- 6. Data Types.

The current version of the Technical Specification has been developed by the Portable Avatar Format Development Committee (PAF-DC). MPAI may issue new versions of MPAI-PAF that extend or replace the scope of the current Technical Specification.

4 Definitions

Capitalised Terms have the meaning defined in <u>*Table 1*</u>. All MPAI-defined Terms are accessible <u>online</u>. Terms with a lower case initial have the meaning commonly defined for the context in which they are used or represent an entity in the real world. For instance,

- 1. Table 1 defines Object, Scene, and User but does not define object, scene, and human.
- 2. Object indicates an Item but object indicates an entity in the Universe commonly classified as object.

A dash "-" preceding a Term in <u>Table 1</u> means the following:

- 1. If the font is normal, the Term in <u>Table 1</u> without a dash and preceding the one with a dash should be placed <u>before</u> that Term. The notation is used to concentrate in one place all the Terms that are composed of, e.g., the word Data <u>followed</u> by one of the words Format and Type.
- 2. If the font is *italic*, the Term in the table without a dash and preceding the one with a dash should be placed <u>after</u> that Term. The notation is used to concentrate in one place all the Terms that are composed of, e.g., the word Descriptor <u>preceded</u> by one of the words Face and Body.

Table 1 – Terms and DefinitionsDefinition

Term

3D Model	Data representing the surface and relevant features of objects from the real world, or Media Data or a mixture of the two.
Avatar	A 3D Model representing a real or fictitious person or Data representing a real person.
- Portable	A Data Type conveying information related to an Avatar such as Avatar ID, Avatar Model, Body Descriptors, Face Descriptors, Speech Data, and Text, and Context information such as Time, Audio-Visual Scene, Spatial Attitude, Language Preference, and Personal Status.
Body	A digital representation of a human body, head included, face excluded.
- Descriptors	A Data Type representing the motion of the body of an Entity.
Face	The digital representation of a human face.
- Descriptors	A Data Type representing the motion and conveying information on the Personal Status of the face of an Entity.
- Identifier	The Identifier of a human inferred from analysing the face of that human.
Gesture	A Data Type representing the movement of the body or part of it, such as the head, arm, hand, and finger, often as a complement to a vocal utterance.
Human	A human being in a real space.
- Digital	A Digitised or a Virtual Human.
- Digitised	Data that has the appearance of a specific human when rendered.
- Virtual	Data created by a computer that has a human appearance when rendered but is not a Digitised Human.
Visual	A Data Type whose instance represents analogue signals – or is rendered to be perceived – in the human-visible range (380 to 700 nanometres).

5 References

5.1 Normative References

Technical Specification: Portable Avatar Format (MPAI-PAF) normatively references the following documents, both from MPAI and other standards organisations. Referenced MPAI standards are publicly available at the URL indicated in the reference.

- 1. MPAI; Technical Specification: Governance of the MPAI Ecosystem (MPAI-GME) V1.1.
- 2. MPAI; Technical Specification: Artificial Intelligence Framework (MPAI-AIF) V2.1.
- 3. MPAI; Technical Specification: Context-based Audio Enhancement (MPAI-CAE) V2.3.
- 4. MPAI; Technical Specification: Object and Scene Description (MPAI-OSD) V1.3.
- 5. MPAI; Technical Specification: <u>AI Module Profiles</u> (MPAI-PRF) V1.0.
- 6. MPAI; Technical Specification: Data Types, Formats, and Attributes; V1.3.
- 7. Khronos; Graphics Language Transmission Format (gITF); October 2021.
- 8. ISO/IEC 19774-1:2019 Information technology Computer graphics, image processing and environmental data representation Part 1: <u>Humanoid animation (HAnim) architecture</u>.
- ISO/IEC 19774-2:2019 Information technology Computer graphics, image processing and environmental data representation – Part 2: <u>Humanoid animation (HAnim) motion data</u> <u>animation</u>.

5.2 Informative References

These references are provided for information purposes.

- 10. MPAI; The MPAI Statutes.
- 11. MPAI; <u>The MPAI Patent Policy</u>.

- 12. MPAI; Technical Specification: <u>Connected Autonomous Vehicles</u> (MPAI-CAV) <u>Technologies</u> (CAV-TEC) V1.0.
- 13. MPAI; Technical Specification: <u>Human and Machine Communication</u> (MPAI-MMM) V2.0.
- 14. MPAI; Technical Specification: MPAI Metaverse Model (MPAI-MMM) Technologies
- V2.0.

6 AI Workflows

Technical Specifications

Technical Specification: Portable Avatar Format (MPAI-PAF) V1.3, jointly with other MPAI Technical Specifications, provides technologies for the digital representation of 3D Model Data that enable the **Avatar-Based Videoconference**, a form of videoconference held in a Virtual Environment populated by speaking Avatars and implemented as an AI Workflow specified by **Technical Specification: AI Framework (MPAI-AIF) V2.1**.

Table 1 displays the full list of AIWs specified by MPAI-PAF V1.3. Click a listed AIW to access its dedicated page, which includes its functions, reference model, I/O Data, Functions of AIMs, I/O Data of AIMs, and a table providing links to the AIW-related AIW, AIMs, and JSON metadata. All previously specified MPAI-PAF AI-Workflows are superseded by those specified by V1.3 but may be used if their version is explicitly mentioned.

Table 1 - AIWs specified by MPAI-PAF V1.4			
Acronym	Names and Specifications of AI Workflows	JSON	
PAF-CTX	Videoconference Client Transmitter	<u>X</u>	
MMC-VMS	Virtual Meeting Secretary	<u>X</u>	
PAF-AVS	Avatar Videoconference Server	<u>X</u>	
PAF-CRX	Videoconference Client Receiver	<u>X</u>	

Figure 1 depicts the system composed of four types of subsystems specified as AI Workflows.



 $Figure \ 1-Avatar\text{-}Based \ Video conference \ end\ to\ end\ diagram$

The components of the PAF-ABV system:

1. **Participant:** a human joining an ABV either individually or as a member of a group of humans in the same physical space.

- 2. Audio-Visual Scene: a Virtual Audio-Visual Environment equipped with Visual Objects such as a Table and an appropriate number of chairs and Audio Objects described by Audio-Visual Scene Descriptors.
- 3. **Portable Avatar:** a data set specified by MPAI-PAF including data representing a human participant.
- 4. Videoconference Client Transmitter:
 - <u>At the beginning</u> of the conference:
 - Receives from Participants and sends to the Server Portable Avatars containing the Avatar Models and Language Selectors.
 - Sends to the Server Speech Object and Face Object for Authentication.
 - <u>Continuously</u> sends to the Server Portable Avatars containing Avatar Descriptors and Speech.

5. The Avatar Videoconference Server

- <u>At the beginning</u> of the conference:
 - Selects the Audio-Visual Descriptors, e.g., a Meeting Room.
 - Equips the Room with Objects, i.e., Table and Chairs.
 - Places Avatar Models around the Table with a given Spatial Attitude.
 - Distributes Portable Avatars containing Avatars Models, their Speech Objects and Spatial Attitudes, and Audio-Visual Scene Descriptors to all Receiving Clients.
 - Authenticates Speech and Face Objects and assigns IDs to Avatars.
 - Sets the common conference language.
- <u>Continuously</u>:
 - Translates Speech to Participants according to their Language Selectors.
 - Sends Portable Avatars containing Avatar Descriptors, Speech, and Spatial Attitude of Participants and Virtual Meeting Secretary to all Receiving Clients and Virtual Meeting Secretary.
- 6. **Virtual Meeting Secretary** is an Avatar not corresponding to any Participant that <u>continuously</u>:
 - Uses the common meeting language.
 - Understands Text Objects and Speech Objects of all Avatars and extracts their Personal Statuses.
 - Drafts a Summary of its understanding of Avatars' Text Objects, Speech Objects, and Personal Status.
 - Displays the Summary either to:
 - Outside of the Virtual Environment for participants to read and edit directly, or
 - The Visual Space for Avatars to comment, e.g., via Text Objects.
 - Refines the Summary.
 - Sends its Portable Avatar containing its Avatar Descriptors to the Server.

7. Videoconference Client Receiver:

- <u>At the beginning</u> of the conference:
 - Receives Audio-Visual Scene Descriptors and Portable Avatars containing Avatar Models with their Spatial Attitudes.
- <u>Continuously</u>:
 - Receives Portable Avatars with Avatar Descriptors and Speech.
 - Produces Visual Scene Descriptors and Audio Scene Descriptors.
 - Renders the Audio-Visual Scene by spatially adding the Avatars' Speech Objects to the Spatial Attitude of the respective Avatars' Mouths. Rendering may be done from a Point of View, possibly different from the Position assigned to their Avatars in the Visual Scene, selected by participant who use a device

of their choice (Head Mounted Display or 2D display/earpad) to experience the Audio-Visual Scene.

Each component of the Avatar-Based Videoconference Use Case is implemented as an AI Workflow (AIW) composed of AI Modules (AIMs). Each AIW includes the following elements:

1	Functions of the AIW	The functions performed by the AIW implementing the Use Case.
2	Reference Model of the AIW	The Topology of AIMs in the AIW.
3	Input and Output Data of the AIW	Input and Output Data of the AIW.
4	Functions of the AIMs	Functions performed by the AIMs.
5	Input and Output Data of the AIW	Input and Output Data of the AIMs.
6	AIW, AIMs, and JSON Metadata	Links to summary specification on the web of the AIMs and corresponding JSON Metadata [2].

Reference Software

As a rule, MPAI provides Reference Software implementing the Technical Specification released with the BSD-3-Clause licence and the following disclaimers

- 1. The purpose of the Reference Software is to demonstrate a working Implementation of an AIW, not to provide a ready-to-use product.
- 2. MPAI disclaims the suitability of the Software for any other purposes than those of the MPAI-PAF Standard and does not guarantee that it offers the best performance and that it is secure.
- 3. Users shall verify that they have the right to use any third-party software required by this Reference Software, e.g., by accepting the licences from third-party repositories.

Note that <u>at this stage</u> only part of the AIMs required to operate the MPAI-PAF AIWs have a Reference Software Implementation.

Conformance Testing

An implementation of an AI Workflow conforms with MPAI-PAF if it accepts as input and produces as output Data and/or Data Objects (Data of a Data Type and its Qualifier) conforming with those specified by MPAI-PAF.

The Conformance of an instance of a Data is to be expressed by a sentence like "Data validates against the Data Type Schema". This means that:

- Any Data Sub-Type is as indicated in the Qualifier.
- The Data Format is indicated by the Qualifier.
- Any File and/or Stream have the Formats indicated by the Qualifier.
- Any Attribute of the Data is of the type or validates against the Schema specified in the Qualifier.

The method to Test the Conformance of a Data or Data Object instance is specified in the *Data Types* chapter.

Performance Assessment

Performance is a multidimensional entity because it can have various connotations. Therefore, the Performance Assessment Specification should provide methods to measure how well an AIW performs its function, using a metric that depends on the nature of the function, such as:

- 1. Quality: the Performance of a <u>Videoconference Client Transmitter</u> AIW can measure how well the AIW represents the human Participant.
- 2. Bias: Performance of a <u>Videoconference Client Receiver</u> AIW can reproduce the avatar videoconference.

- 3. Legal compliance: the Performance of an AIW can measure the compliance of the AIW to a regulation, e.g., the European AI Act.
- 4. Ethical compliance: the Performance Assessment of an AIW can measure the compliance of an AIW to a target ethical standard.

Note that at this stage MPAI-PAF AIWs do not have a Performance Assessment Specification.

6.1 Videoconference Client Transmitter

6.1.1 Functions

The functions of the Videoconference Client Transmitter are to:

- 1. <u>Receive</u> from a Participant:
 - Input Audio from the microphone.
 - Input Visual from the camera.
 - Participant's Avatar Model.
 - Participant's Language Selector (e.g., EN-US, IT-CH).
- 2. <u>Send</u> to the Server:
 - Speech Object (for Authentication).
 - Face Object (for Authentication).
 - Input Portable Avatars containing:
 - Language Selector (at the start).
 - Avatar Model (at the start).
 - Input Speech.
 - Avatar Descriptors.

6.1.2 Reference Model

Figure 1 gives the Reference Model of Client Transmitter AIW. Red text refers to data sent at meeting start.



Figure 1 – Reference Model of Videoconference Client Transmitter (PAF-ABV)

At the start, each participant provides:

- 1. Language Selector
- 2. Avatar Model.
- 3. Speech Object (for Authentication).
- 4. Face Object (for Authentication).
- 5. Participant ID
- During the videoconference:
- 1. Audio-Visual Scene Description produces Speech Objects, Face Objects, Face Descriptors, Body Descriptors and Audio-Visual Scene Geometry.
- 2. Automatic Speech Recognition produces Recognised Text.
- 3. Personal Status Extraction produces Personal Status.
- 4. Portable Avatar Multiplexing multiplexes Recognised Text, Personal Status, Input Speech, Face Descriptors, Body Descriptors, Language Selector, Avatar Model, and Participant ID.
- 5. Videoconference Client Transmitter sends Portable Avatars to Avatar Videoconference Server that the Server processes and re-distributes to Client Receivers.

6.1.3 Input and Output Data

Table 1 gives the input and output data of the Client Transmitter AIW:

Table 1 – Input and output data of Client Transmitter AIW

Input	Description
Input <u>Text</u>	Chat text used by a participant to communicate with Virtual Meeting Secretary or other participants
input Selector	The language(s) a participant wishes to speak and hear.
Input <u>Audio</u>	Audio of a participants' Speech in a meeting room.
Input <u>Visual</u>	Visual of participants in a meeting room.
Avatar <u>Model</u>	The avatar model selected by the participant.
Output	Description
Speech Object	A participant's utterance used by Server for authentication.
Participant Portable Avatar	Portable Avatar produced by Client Transmitter.
Face Object	Participant's face used by Server for authentication.

6.1.4 Functions of AI Modules

Table 2 gives the functions of AI Modules of the Client Transmitter AIW.

Table 2 – AT Modules of Client Transmitter ATW		
AIM	Function	
Audio-Visual Scene Description	 Receives Input Audio and Input Visual. Provides Input Speech, Speech Object, Face Descriptors, Body Descriptors, Face Object. 	
Automatic Speech	1. Receives Input Speech.	
Recognition	2. Provides Recognised Text.	

Table 2 AI Modules of Client Trans

	1. Receives Recognised Text, Speech, Face Descriptors, Body
Personal Status Extraction	Descriptors.
	2. Provides the Participant's Personal Status.
	1. Receives Language Selector, Avatar Model, Input Text, Input
Portable Avatar	Speech, Recognised Text, Personal Status, Participant ID, Face
<u>Multiplexing</u>	Descriptors, Body Descriptors.
	2. Provides Participant Portable Avatars.

6.1.5 I/O Data of AI Modules

Table 3 gives the AI Modules of Client Transmitter AIW.

AIM	Input	Output
Audio-Visual Scene Description	Input <u>Audio</u> Input <u>Visual</u>	Input <u>Speech</u> <u>Speech Object</u> Face <u>Object</u> <u>Face Descriptors</u> <u>Body Descriptors</u>
Automatic Speech Recognition	Input Speech	Recognised Text
Personal Status Extraction	Recognised <u>Text</u> Input <u>Speech</u> <u>Face Descriptors</u> <u>Body Descriptors</u>	Personal Status
Portable Avatar Multiplexing	Recognised <u>Text</u> <u>Personal Status</u> Input <u>Speech</u> <u>Face Descriptors</u> <u>Body Descriptors</u> Input <u>Text</u> Language <u>Selector</u> Avatar <u>Model</u> Participant ID	<u>Portable Avatar</u> .

Table 3 – AI Modules of Client Transmitter AIW

6.1.6 AIW, AIM, and JSON Metadata

AIW	AIMs/1	AIMs/2	AIMs/3	Name	JSON
PAF-CTX				Videoconference Client Transmitter	<u>X</u>
	OSD-AVS			Audio-Visual Scene Description	<u>X</u>
		CAE-ASD		Audio Scene Description	<u>X</u>
			CAE-AAT	Audio Analysis Transform	<u>X</u>
			CAE-ASL	Audio Source Localisation	<u>X</u>

		CAE-ASE	Audio Separation and Enhancement	<u>X</u>
		CAE-AST	Audio Synthesis Transform	<u>X</u>
		CAE-AMX	Audio Descriptors Multiplexing	<u>X</u>
	OSD-VSD		Visual Scene Description	<u>X</u>
	OSD-AVA		Audio-Visual Alignment	<u>X</u>
MMC-ASR			Automatic Speech Recognition	<u>X</u>
MMC-PSE			Personal Status Extraction	<u>X</u>
	MMC-ETD		Entity Text Description	<u>X</u>
	MMC-ESD		Entity Speech Description	<u>X</u>
	PAF-EFD		Entity Face Description	<u>X</u>
	PAF-EBD		Entity Body Description	<u>X</u>
	MMC-PTI		PS-Text Interpretation	<u>X</u>
	MMC-PSI		PS-Speech Interpretation	<u>X</u>
	PAF-PFI		PS-Face Interpretation	<u>X</u>
	PAF-PGI		PS-Gesture Interpretation	<u>X</u>
	MMC-PMX		Personal Status Multiplexing	<u>X</u>
PAF-PMX			Portable Avatar Multiplexing	X

6.1.7 Conformance Testing

Table 2 provides the Conformance Testing Method for PAF-CTX AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 - Conformance Testing Method for PAF-CTX AIM

Receives Input Text	Shall validate against Text Object Schema. Text Data shall conform with Text Qualifier.
input Selector	Shall validate against Selector Schema.
Input <u>Audio</u>	Shall validate against Audio Object Schema. Speech Data shall conform with Audio Qualifier.
Input Visual	Shall validate against Visual Object Schema. Visual Data shall conform with Visual Qualifier.
Avatar Model	Shall validate against 3D Model Object Schema. Speech Data shall conform with 3D Model Qualifier.
Produces Speech Object	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.

Participant <u>Portable</u> <u>Avatar</u>	Shall validate against Portable Avatar Schema. Portable Avatar Data shall conform with respective Qualifiers.	
Face Object	Shall validate against Visual Object Schema. Face Data shall conform with Visual Qualifier.	

7 AI Modules

Technical Specifications

Table 1 provides the links to the specifications and the JSON syntax of all AIMs specified by *Technical Specification: Portable Avatar Format (MPAI-PAF) V1.4*. MPAI-PAF V1.4 AI-Modules supersede those previously specified which may still be used if their version is by explicitly signalled. AIMs in bold are Composite.

Table 1 – Specifications and JSON syntax of AIMs used by MPAI-PAF V1.4

Acronym	AIM Name	JSON	Acronym	AIM Name	JSON
PAF- AVC	Audio-Visual Scene Creation	<u>X</u>	PAF-PDX	Portable Avatar Demultiplexing	<u>X</u>
PAF- AVR	Audio-Visual Scene Rendering	X	PAF- PMX	Portable Avatar Multiplexing	<u>X</u>
PAF-EBD	Entity Body Description	<u>X</u>	PAF-PFI	PS-Face Interpretation	<u>X</u>
PAF-EFD	Entity Face Description	<u>X</u>	PAF-PGI	PS-Gesture Interpretation	<u>X</u>
PAF-FIR	Face Identity Recognition	<u>X</u>	PAF-PSD	Personal Status Display	<u>X</u>
PAF-FPS	<u>Face Personal Status</u> <u>Extraction</u>	<u>X</u>	PAF-SPA	Service Participant Authentication	<u>X</u>
PAF- GPS	Gesture Personal Status Extraction	<u>X</u>	PAF-VSC	Visual Scene Creation	<u>X</u>

Conformance Testing

An implementation of an AI Module conforms with MPAI-PAF V1.4 if it accepts as input and produces as output Data and/or Data Objects (combination of Data of a Data Type and its Qualifier) conforming with those specified by MPAI-PAF V1.4.

The Conformance of an instance of a Data is to be expressed by a sentence like "Data validates against the Data Type Schema". This means that:

- Any Data Sub-Type is as indicated in the Qualifier.
- The Data Format is indicated by the Qualifier.
- Any File and/or Stream have the Formats indicated by the Qualifier.
- Any Attribute of the Data is of the type or validates against the Schema specified in the Qualifier.

The method to Test the Conformance of a Data or Data Object instance is specified in the *Data Types* chapter.

Performance Assessment

Performance is a multidimensional entity because it can have various connotations. Therefore, the Performance Assessment Specification should provide methods to measure how well an AIW performs its function, using a metric that depends on the nature of the function, such as:

- 1. *Quality*: Performance Assessment measures how well an AIM performs its function, using a metric that depends on the nature of the function, e.g., how well a Face Identity Recognition (FIR) AIM identifies Faces.
- 2. *Bias*: Performance Assessment measures the preference given by an AIM to certain elements, using a metric that depends on a bias related to certain attributes of the AIM. For instance, a Face Identity Recognition (FIR) AIM tends to have a higher correct identification of Face having a particular skin colour.
- 3. *Legal* compliance: Performance Assessment measures how well an AIM performs its function, using a metric that assesses its accordance with a certain legal standard.
- 4. *Ethical* compliance: the Performance Assessment of an AIM can measure the compliance of an AIM to a target ethical standard.

The current MPAI-PAF V1.4 does not provide AIM Performance Assessment methods.

7.1 Audio-Visual Scene Creation

7.1.1 Functions

Audio-Visual Scene Creation (PAF-AVC) V1.4 provides the Descriptors of an Audio-Visual Scene where the Speech Objects associated with each provided Avatars are attached to the Mouth of each Avatar using the Mouths' Spatial Attitudes:

Receives	Visual Scene Descriptors	Descriptors of a Visual Scene.
	Avatars	Avatars in the Visual Scene.
	Mouth Spatial Attitude	Spatial Attitudes of the Avatars in the Scene.
	Speech Objects	Speech Objects uttered by the Avatars
Produces	Audio-Visual Scene Descriptors	New AV Scene Descriptors where Speech Objects
110ddees 11ddio 11sudi Seene Des	Hudio Histali Seene Desemptors	are located at Avatars' mouths.

7.1.2 Reference Model

Figure 1 depicts the Reference Model of the Audio-Visual Scene Creation (PAF-AVC) AIM.



Figure 1 Audio-Visual Scene Creation (PAF-AVC) AIM

7.1.3 Input/Output Data

Table 1 specifies the Input and Output Data of Audio-Visual Scene Creation (PAF-AVC) AIM.

Table 1 – I/O Data of the Audio-Visual Scene Creation (PAF-AVC) AIM

Input	Description
Visual Scene Descriptors	Input Visual Scene Descriptors
Mouth Spatial Attitudes	Spatial Attitude of Avatars' Mouths
Avatars	IDs of Avatars
Speech Objects	Speech Objects associated to Avatars
Output	Description
Audio-Visual Scene Descript	ors Output AV Scene Descriptors

7.1.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.4/AIMs/AudioVisualSceneCreation.json

7.1.5 Conformance Testing

Table 2 provides the Conformance Testing Method for the PAF-AVC AIM as a Basic AIM.

Note that a schema may contain references to other schemas. In this case, validation of data for the primary schema implies that any data that refers to a secondary schema shall also validate.

Table 2 - Conformance Testing Method for PAF-AVC AIM

Receives	Visual Scene Descriptors	Shall validate against Visual Scene Descriptors Schema.
	Mouth Spatial Attitudes	Shall validate against Spatial Attitude Schema.
	<u>Avatar</u> s	Shall validate against Avatar Schema. Avatar Model Data shall conform with 3D Model Qualifier.
	Speech Objects	Shall validate against Avatar Schema. Speech Data shall conform with Speech Qualifier.
Produces	Audio-Visual Scene Descriptors	Shall validate against Visual Scene Descriptors Schema.

7.2 Audio-Visual Scene Rendering

7.2.1 Functions

Audio-Visual Scene Rendering (PAF-AVR) produces Speech, Audio, and Visual Objects from a Portable Avatar, Audio-Visual Scene Descriptors and a Point of View:

Receives Point of View To be used in rendering the scene and its objects.

	Audio-Visual Scene Descriptors	jointly with or alternatively with Portable Avatar.
	Portable Avatar	Jointly with or alternatively with AV Scene Descriptors.
Transforms	Portable Avatar	Into generic Audio-Visual Scene Descriptors if input is Portable Avatar.
Produces	Output Speech	Of Portable Avatar integrated in the Audio-Visual Scene. Output Speech results from the rendering of Audio Scene Descriptors from human-selected Point of View.
	Output Visual	Resulting from the rendering of Audio Scene Descriptors from human-selected Point of View. View Selector tells the OSD-AVR AIM where the visual components of the Portable Avatar should be integrated.

7.2.2 Reference Model

Figure 1 specifies the Reference Model of the Audio-Visual Scene Rendering (PAF-AVR) AIM.



Figure 1 – The Audio-Visual Scene Rendering (PAF-AVR) AIM

7.2.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio-Visual Scene Rendering (PAF-AVR) AIM.

Table 1 – I/O Data of the Audio-Visual Scene Rendering (PAF-AVR) AIM

Input	Description
Portable Avatar	Data produced, e.g., by Personal Status Display.
AV Scene Descriptors	Audio-Visual Scene Descriptors.
Point of View	Point from where an Entity perceives the Audio-Visual Scene
Output	Description
Output Speech Object	The Speech components of the Audio-Visual Scene.
Output Audio Object	The Audio components of the Audio-Visual Scene.
Output Wiscal Ohis at	

7.2.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.4/AIMs/AudioVisualSceneRendering.json

7.2.5 Profiles

The Profiles of Audio-Visual Scene Rendering are specified.

7.2.6 Conformance Testing

Table 2 provides the Conformance Testing Method for PAF-AVR AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for PAF-AVR AIM

Receives	Portable Avatar	Shall validate against Point of View Schema.
	AV Scene Descriptors	Shall validate against AV Scene Descriptors Schema.
	Point of View	Shall validate against Portable Avatar Schema. Portable Avatar Data shall conform with respective Qualifiers.
Produces	Output Speech Object	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.
	Output Audio Object	Shall validate against Audio Object Schema. Audio Data shall conform with Audio Qualifier.
	Output Visual Object	Shall validate against Visual Object or 3D Model Schema. Visual Data shall conform with Visual Object.

7.3 Entity Body Description

7.3.1 Functions

Entity Body Description (PAF-EBD) V1.4 produces the Body Descriptors of a Visual Object representing a human body:

Receives *Body Visual Object* Body of Entity or from upstream AIM. Produces *Body Descriptors* Descriptors of Body Visual Object

7.3.2 Reference Model

Figure 1 specifies the Reference Model of the Entity Body Description (PAF-EBD) AIM.



Figure 1 Entity Body Description (PAF-EBD) AIM

7.3.3 Input/Output Data

Table 1 specifies the Input and Output Data of Entity Body Description (PAF-EBD) AIM.

Table 1 – I/O Data of the Entity Body Description (PAF-EBD) AIM

InputDescriptionBody Visual ObjectVisual Object representing the body of an Entity.

Description

Output

Body Descriptors Body Descriptors of Visual Object.

7.3.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.4/AIMs/EntityBodyDescription.json

7.3.5 Conformance Testing

Table 2 provides the Conformance Testing Method for MMC-EBD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 - Conformance Testing Method for MMC-EBD AIM

Receives Body Visual ObjectShall validate against Visual Object Schema.Body Data shall conform with Visual Object.Produces Body DescriptorsShall validate against Body Descriptors XML Schema.

7.4 Entity Face Description

7.4.1 Functions

Entity Face Description (PAF-EFD):

Receives *Face Visual Object* Face of Entity or from upstream AIM. Produces *Face Descriptors Descriptors of Entity Face*.

7.4.2 Reference Model

Figure 1 specifies the Reference Model of the Entity Face Description (PAF-EFD) AIM.



Figure 1 - Entity Face Description (PAF-EFD) AIM

7.4.3 Input/Output Data

Table 1 specifies the Input and Output Data of Entity Face Description (PAF-EFD) AIM.

Table 1 – I/O Data of the Entity Face Description (PAF-EFD) AIM

Input	Description
Face Visual Object	t Entity Face to be Described
Output	Description
Face Descriptors	Descriptors of Face.

7.4.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.4/AIMs/EntityFaceDescription.json

7.4.5 Reference Software

The open-source Reference Software is <u>available</u>. Send an email to the <u>MPAI Secretariat</u> to access the code.

7.4.6 Conformance Testing

Table 2 provides the Conformance Testing Method for PAF-EFD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for PAF-EFD AIM

Receives Face Visual Object	Shall validate against Visual Object Schema. Face Data shall conform with Visual Qualifier.
Produces Face Descriptors	Shall validate against Face Descriptors Schema.

7.5 Face Identity Recognition

7.5.1 Functions

Face Identity Recognition (PAF-FIR) produces the Bounding Box with the face and the identity of the face from the image and the geometry of the Visual Scene the Image Visual Object it belongs to:

Receives	Text Object	Text that is related with the Face to be identified.
	Image Visual Object	Image containing Face to be identified.
	Face Time	Time when the face should be identified.
	Visual Scene Geometr	ry Of the scene where the Face is located.
Searches for	or Bounding Boxes	That include faces
Finds	best match	Between the Faces and those in a database.
Produces	Face Identities	Face Instance Identifiers.
	Bounding Boxes	Bounding Boxes that include faces.

7.5.2 Reference Model

Figure 1 depicts the Reference Model of the Face Identity Recognition AIM.



Figure 1 – Face Identity Recognition AIM

7.5.3 Input/Output Data

Table 1 specifies the Input and Output Data of the of the Face Identity Recognition AIM.

Table 1 –	I/O Data	of the	Face	Identity	Recognition	AIM
-----------	----------	--------	------	----------	-------------	-----

Input	Description
Auxiliary <u>Text Objext</u>	Text with a content related to Face ID.
Image Visual Object	An image containing the Face to be identified.
Face <u>Time</u>	The Time during which the Face should be identified.
Visual Scene Geometry	The Geometry of the Scene where the Face is located.
Output	Description

Face Identifiers	Associate strings to elements belonging to some levels in a hierarchical classification (taxonomy).
Bounding Boxes	The box containing the Face identified.

7.5.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.4/AIMs/FaceIdentityRecognition.json

7.5.5 Reference Software

7.5.5.1 Disclaimers

- 1. This PAF-FIR Reference Software Implementation is released with the BSD-3-Clause licence.
- 2. The purpose of this PAF-FIR Reference Software is to show a working Implementation of PAF-FIR, not to provide a ready-to-use product.
- 3. MPAI disclaims the suitability of the Software for any other purposes and does not guarantee that it is secure.
- 4. Use of this Reference Software may require acceptance of licences from the respective repositories. Users shall verify that they have the right to use any third-party software required by this Reference Software.

7.5.5.2 Guide to the PAF-FIR code

Use of this Reference Software for the PAF-FIR AI Module is for developers who are familiar with Python, Docker, RabbitMQ, and downloading models from HuggingFace

PAF-FIR performs face identity recognition with a pretrained FaceNet model; that is, it identifies the faces in a given number of frames per scene by comparison with a dataset of faces.

The PAF-FIR Reference Software is found at the MPAI gitlab site. It contains:

- 1. src: a folder with the Python code implementing the AIM
- 2. Dockerfile: a Docker file containing only the libraries required to build the Docker image and run the container
- 3. requirements.txt: dependencies installed in the Docker image
- 4. README.md: where to find and save weights of face recognition model FaceNet512.

Library: https://github.com/serengil/deepface

7.5.5.3 Acknowledgements

This version of the PAF-FIR Reference Software has been developed by the MPAI *AI Framework* Development Committee (AIF-DC).

7.5.6 Conformance Testing

Table 2 provides the Conformance Testing Method for PAF-FIR AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for PAF-FIR AIM

Receives	Text Object	Shall validate against Text Object Schema.
	Visual Object (Image)	Shall validate against Visual Object Schema. Image Data shall conform with Visual Qualifier.
	Face <u>Time</u>	Shall validate against Time Schema.
	Visual Scene Geometry	Shall validate against Visual Scene Geometry Schema.
Produces	Face Instance IDs	Shall validate against Instance ID Schema.
	<u>Visual Object</u> (Bounding Box)	Shall validate against Bounding Box Schema. Bounding Box Data shall conform with Visual Qualifier.

7.5.7 Performance Assessment

Performance Assessment of a PAF-FIR AIM Implementation shall be performed using a dataset of faces for each face of which the Identity of the face is provided with reference to a Taxonomy.

The Performance Assessment Report of an PAF-FIR AIM Implementation shall include:

- 1. The Identifier of the PAF-FIR AIM.
- 2. The identifier of the face dataset.
- 3. The identifier of the Taxonomy of face identifiers.
- 4. The Performance of the PAF-FIR AIM Implementation expressed by the Accuracy of the Identifiers provided by the output of the PAF-FIR AIM computed on all faces of the dataset referenced in 2 using the Taxonomy referenced in 3.

7.6 Face Personal Status Extraction

7.6.1 Functions

Face Personal Status Extraction (PAF-FPE) extracts the Personal Status of a Face Visual Object:

Receives Input Selector	Signalling whether a Face Object or Face Descriptors are provided.
Face Object	From which the Personal Status should be extracted.
Face Descriptors	From which the Personal Status should be extracted (Externally computed).
Produces Face Personal Status	The Personal Status of the Face Visual Object

7.6.2 Reference Model

Figure 1 specifies the Reference Model of the Face Personal Status Extraction (PAF-FPE) AIM.



Figure 1 – Reference Architecture of the Face Personal Status Extraction (PAF-FPE) AIM

7.6.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Face Personal Status Extraction (PAF-FPE) AIM.

Table 1 – I/O Data of the Face Personal Status Extraction (PAF-FPE) AIM

Input	Description
Input <u>Selector</u>	Signals whether a Face Object or its Face Descriptors should be used to extracts Face Personal Status
Face Visual Object	Face Object from which the Personal Status should be extracted.
Face Descriptors	Descriptors from which the Personal Status should be extracted.
Output	Description
Face Personal Status	The computed Face Personal Status.

7.6.4 SubAIMs

A Face Personal Status Extraction (PAF-FPE) AIM instance may be implemented as a Composite AIM as specified in Figure 2.

Input Face Selection	Entity Faco	Face Descriptors]
Face Object	Description		PS-Face -	Face
Face Descriptors]	Interpretation	
		/		

Figure 2 - Reference Model of Face Personal Status Extraction (PAF-FPE) Composite AIM

The AIMs composing the Face Personal Status Extraction (PAF-FPE) Composite AIM are:

AIM	AIMs	Names	JSON
PAF-FPE		Face Personal Status Extraction	<u>X</u>
	PAF-EFD	Entity Face Description	<u>X</u>
	PAF_PFI	PS-Face Interpretation	<u>X</u>

7.6.5 JSON Metadata

https://schemas.mpai.community/PAF/V1.4/AIMs/FacePersonalStatusExtraction.json

7.6.6 Conformance Testing

Table 2 provides the Conformance Testing Method for PAF-FPE AIM. Conformance Testing of the individual AIMs of the PAF-FPE Composite AIM are given by the individual AIM Specification.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for PAF-FPE AIM

Receives Input SelectorShall validate against Selector Schema.Visual Object (Face)Shall validate against Face Object Schema.Face DescriptorsFace Data shall conform with Visual Qualifier.Face Personal StatusShall validate against Face Personal Status Schema.

Table 3 provides an example of MMC-FPE AIM conformance testing. This specification provides the methods and the datasets to test the conformance of a PAF-FPE instance without input Face Descriptors.

Table 3 - An example MMC-FPE AIM conformance testing

Input Data	Data Type	Input Conformance Testing Data
Face Object	AVC	All input Video files to be drawn from Video files.
Output Data	Data Type	Data Format
Emotion (Face)	JSON	All Emotion JSON Files shall validate against <u>Emotion</u> JSON Schema.

emotion_Name and emotion_SetName must be present in the output JSON file of Emotion. The value of either of the two may be null.

7.7 Gesture Personal Status Extraction

7.7.1 Functions

Gesture Personal Status Extraction (PAF-GPE) produces the Gesture Personal Status of a Body Image Object:

Receives Input Selector	Signalling whether a Gesture Object or Gesture Descriptors are provided.
Body Visual Object	From which the Personal Status should be extracted.

Gesture Descriptors	From which the Personal Status should be extracted
	(Externally computed).
Produces Gesture Personal Status.	The Gesture Personal Status of the Body Image Object.

7.7.2 Reference Model

Figure 1 specifies the Reference Model of the Gesture Personal Status Extraction (PAF-GPE) AIM.



Figure 1 - Reference Architecture of the Gesture Personal Status Extraction (PAF-GPE) AIM

7.7.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Gesture Personal Status Extraction (PAF-GPE) AIM.

Table 1 – I/O Data of the Gesture Personal Status Extraction (PAF-GPE) AIM

Input	Description
Input <u>Selector</u>	Signals whether a Body Object or its Gesture Descriptors should be used to extracts Gesture Personal Status.
Body Visual Object	Gesture Object from which the Personal Status should be extracted.
Gesture Descriptors	Descriptors from which the Personal Status should be extracted
Output	Description

Gesture Personal Status The computed Gesture Personal Status.

7.7.4 SubAIMs

A Gesture Personal Status Extraction (PAF-GPE) AIM instance may be implemented as a Composite AIM as specified in Figure 2.



Figure 2 - Reference Model of Gesture Personal Status Extraction (PAF-GPE) Composite AIM

The AIMs composing the Gesture Personal Status Extraction (PAF-GPE) Composite AIM are:

AIM	AIMs	Names	JSON
PAF-GPE		Gesture Personal Status Extraction	<u>X</u>
	PAF-EGD	Entity Gesture Description	<u>X</u>
	PAF-PGI	PS-Gesture Interpretation	<u>X</u>

7.7.5 JSON Metadata

https://schemas.mpai.community/PAF/V1.4/AIMs/GesturePersonalStatusExtraction.json

7.7.6 Conformance Testing

Table 2 provides the Conformance Testing Method for PAF-GPE AIM. Conformance Testing of the individual AIMs of the PAF-GPE Composite AIM are given by the individual AIM Specification.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for PAF-GPE AIM

Receives Input <u>Selector</u>	Shall validate against Selector Schema.
Visual Object (Body)	Shall validate against Visual Object Schema. Body Data shall conform with Visual Object.
Gesture Descriptors	Shall validate against Body Descriptors Schema.
Produces Gesture Personal Status	Shall validate against Body Personal Status Schema.

7.8 Portable Avatar Demultiplexing

7.8.1 Functions

Portable Avatar Demultiplexing (PAF-PDX) V1.4 extracts the components of a Portable Avatar:

Receives Portable Avatar

Demultiplexes Elements in Portable Avatar.

Produces - Portable Avatar ID

- Avatar Space-Time
- Avatar
- Language Selector
- Speech Object
- Text Object
- Speech Model

- Personal Status
- Audio Visual Scene Descriptors
- Audio Visual Scene Space Time

7.8.2 Reference Model

Figure 1 specifies the Reference Model of the Personal Avatar Demultiplexing (PAF-PDX) AIM.



Figure 1- The Personal Avatar Demultiplexing (PAF-PDX) AIM

7.8.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Portable Avatar Demultiplexing (PAF-PDX) AIM.

Input	Description
Portable Avatar	From an upstream AIM or another AIW.
Output	Description
AvatarID	Avatar ID.
Avatar Space-Time	Portable Avatar Time.
Avatar	Avatar in Portable Avatar.
Language Selector	Language of Avatar.
Speech Object	The Speech in the time when the PA is valid.
Text Object	The Time in the time when the PA is valid.
Speech <u>Model</u>	The NN Model used to synthesise text.
Avatar Personal Status	The Avatar's Personal Status.
AV Scene Descriptors	Descriptors of AV Scene.

Table 1 –	Portable Avata	r Demultinlexing	(PAF-PDX)	AIM
	T OITADIC AVAIA	Demuniplexing	(I AI - I DA)	

AV Scene Space-Time Space-Time info of AV Scene.

7.8.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.4/AIMs/PortableAvatarDemultiplexing.json

7.8.5 Conformance Testing

Table 2 provides the Conformance Testing Method for PAF-PDX AIM.

Note that a schema may contain references to other schemas. In this case, validation of data for the primary schema implies that any data that refers to a secondary schema shall also validate.

Table 2 - Conformance Testing Method for PAF-PDX AIM

Receives Portable Avatar	Shall validate against Portable Avatar Schema. Portable Avatar Data shall conform with respective Qualifiers.
Produces Portable Avatar ID	Shall be string or validate against Instance ID Schema.
Avatar Space-Time	Shall validate against Space-Time Schema.
Avatar	Shall validate against Avatar Schema. Avatar Model Data shall conform with 3D Model Qualifier.
Language <u>Selector</u>	Shall validate against "Language" Selector Schema.
Text Object	Shall validate against Text Object Schema. Text Data shall conform with Text Qualifier.
Speech Object	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.
Speech Model	Shall validate against Machine Learning Model Schema. Speech Model Data shall conform with Machine Learning Model Qualifier.
Personal Status	Shall validate against Personal Status Schema.
Audio Visual Scene Descriptors	Shall validate against AV Scene Descriptors Schema.
Audio Visual Scene Space-Time	Shall validate against Space-Time Schema.

7.9 Portable Avatar Multiplexing

7.9.1 Functions

Portable Avatar Multiplexing (PAF-PMX) V1.4 integrates the component elements of a Portable Avatar:

Receives An arbitrary number of elements in Portable Avatar out of:

- Portable Avatar ID

- Avatar Space-Time
- Avatar
- Language Selector
- Text Object
- Speech Model
- Speech Object
- Personal Status
- Audio- Visual Scene Space-Time
- Audio-Visual Scene Descriptors
- An existing Portable Avatar

Changes Existing with Input Data

Adds Input Data that is not in the Input Portable Avatar

Produces Portable Avatar

7.9.2 Reference Model

Figure 1 specifies the Reference Model of the Portable Avatar Multiplexing (PAF-PMX) AIM.



Figure 1 – The Portable Avatar Multiplexing (PAF-PMX) AIM

7.9.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Portable Avatar Multiplexing (PAF-PMX) AIM.

Table 1 – I/O Data of the Portable Avatar Multiplexing (PAF-PMX) AIM

Input Des

Description

AvatarID	Avatar ID.
Avatar Space-Time	Portable Avatar Time.
Avatar	Avatar in Portable Avatar.
Language Selector	Language of Avatar.
Speech Object	The Speech in the time when the PA is valid.
Text Object	The Time in the time when the PA is valid.
Speech <u>Model</u>	The NN Model used to synthesise text.
Avatar Personal Status	The Avatar's Personal Status.
AV Scene Descriptors	Descriptors of AV Scene.
AV Scene Space-Time	Space-Time info of AV Scene.
Portable Avatar	The input Portable Item.
Output	Description
Portable Avatar	The output Portable Item.

7.9.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.4/AIMs/PortableAvatarMultiplexing.json

7.9.5 Conformance Testing

Table 2 provides the Conformance Testing Method for PAF-PMX AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for PAF-PMX AIM

Receives	Portable Avatar ID	Shall be string or validate against Instance ID Schema
	Avatar Space-Time	Shall validate against Space-Time Schema
	<u>Avatar</u>	Shall validate against Avatar Schema. Avatar Model Data shall conform with 3D Model Qualifier.
	Language Selector	Shall validate against Selector Schema
	Text Object	Shall validate against Text Object Schema. Text Data shall conform with Text Qualifier.
	Speech Object	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.
	Speech <u>Model</u>	Shall validate against Machine Learning Model Schema. Speech Model Data shall conform with Machine Learning Model Qualifier.
	Personal Status	Shall validate against Personal Status Schema.

	Audio Visual Scene Descriptors	Shall validate against AV Scene Descriptors Schema.
	Audio Visual Scene Space-Time	Shall validate against Space-Time Schema.
Produces	Portable Avatar	Shall validate against Portable Avatar Schema. Portable Avatar Data shall conform with respective Qualifiers.

7.10 PS-Face Interpretation

7.10.1 Functions

PS-Face Interpretation (PAF-PFI):

Receives *Face Descriptors* from Face Description or as input to PS-Face Interpretation Produces *Face Personal Status* the Personal Status of the Face Modality

7.10.2 Reference Model

Figure 1 specifies the Reference Architecture of the PS-Face Interpretation (PAF-PFI) AIM.



Figure 1- The PS-Face Interpretation (PAF-PFI) AIM Reference Model

7.10.3 Input/Output Data

Table 1specifies the Input and Output Data of the PS-Face Interpretation (PAF-PFI) AIM.

Table 1– I/O Data of the PS-Face Interpretation (PAF-PFI) AIM

InputDescriptionFace DescriptorsDescriptors of FaceOutputDescriptionFace Personal StatusPersonal Status of Face

7.10.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.4/AIMs/PSFaceInterpretation.json

7.10.5 Conformance Testing

Table 2 provides the Conformance Testing Method for PAF-PFI AIM.

Note that a schema may contain references to other schemas. In this case, validation of data for the primary schema implies that any data that refers to a secondary schema shall also validate.

Table 2 – Conformance Testing Method for PAF-PFI AIM

Receives Face DescriptorsShall validate against Face Descriptors SchemaProduces Face Personal StatusShall validate against Face Personal Status Schema

7.11 PS-Gesture Interpretation

7.11.1 Functions

PS-Gesture Interpretation (PAF-PGI):

ReceivesGesture Descriptorsfrom Gesture Description or as input to PS-Gesture
InterpretationProducesGesture Personal Statusthe Personal Status of the Gesture Modality

7.11.2 Reference Model

Figure 1 specifies the Reference Architecture of the PS-Gesture Interpretation (PAF-PGI) AIM.



Figure 1– The PS-Gesture Interpretation (PAF-PGI) AIM Reference Model

7.11.3 Input/Output Data

Table 1specifies the Input and Output Data of the PS-Gesture Interpretation (PAF-PGI) AIM.

Table 1– I/O Data of the PS-Gesture Interpretation (PAF-PGI) AIM

Input	Description
Gesture Descriptors	Descriptors of Gesture
Output	Description
Gesture Personal Status Personal Status of Gesture	

7.11.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.4/AIMs/PSGestureInterpretation.json

7.11.5 Conformance Testing

Table 2 provides the Conformance Testing Method for PAF-PGI AIM.

Note that a schema may contain references to other schemas. In this case, validation of data for the primary schema implies that any data that refers to a secondary schema shall also validate.

Table 2 - Conformance Testing Method for PAF-PGI AIM

Receives Gesture DescriptorsShall validate against Gesture Descriptors SchemaProduces Gesture Personal StatusShall validate against Gesture Personal Status Schema

7.12 Personal Status Display

7.12.1 Functions

Personal Status Display (PAF-PSD) V1.4 produces the Portable Avatar corresponding to an Avatar Model speaking a Text Object synthesised with a Speech Model and displaying a Personal Status:

Receives Machine IDID to be used to identify the Avatar in Portable Avatar.Text ObjectText associated to Avatar in Portable Avatar.Personal StatusPersonal Status associated to Avatar in Portable Avatar.Avatar Model3D Model associated to Avatar in Portable Avatar.Speech ModelSpeech Model Associated to Avatar in Portable Avatar.Produces Portable AvatarOutput Portable Avatar.EnablesPAF-AVRTo render the Portable Avatar produced by PAF-PSD.

7.12.2 Reference Model

Figure 1 depicts the AIMs implementing the Personal Status Display (PAF-PSD) AIM.





7.12.3 Input/Output Data

Table 1 gives the Input/Output Data of Personal Status Display (PAF-PSD).

Table 1 – I/O Data of Personal Status Display

Input data	From	Description
Avatar ID	Upstream AIM	Portable Avatar's ID
Avatar <u>Model</u>	Upstream AIM or embedded in PSD	Part of Portable Avatar
Text Object	Keyboard or upstream AIM	Texts of Portable Avatar
Personal Status	Personal Status Extraction or Machine	To add PS to Speech, Face, and Gesture
Speech Model	Upstream AIM or embedded in PSD	Neural Network
Output data	То	Description
Portable Avatar	Downstream AIM or renderer	As Portable Avatar

7.12.4 SubAIMs

Figure 2 gives the Reference Model of the the Personal Status Display Composite AIM.



Figure 2 – Reference Model of Personal Status Display Composite AIM

The Personal Status Display Composite AIM operates as follows:

- 1. Avatar ID is the ID of the Portable Avatar.
- 2. Personal Status Demultiplexing makes available the component PS-Speech, PS-Face, and PS-Gesture Modalities.
- 3. Machine Text is synthesised as Speech using a Speech Model in a format specified by NN Format and the Personal Status provided by PS-Speech.
- 4. Machine Speech and PS-Face are used to produce the Machine Face Descriptors.
- 5. PS-Gesture and Text are used for Machine Body Descriptors using the Avatar Model.
- 6. Portable Avatar Multiplexing produces the Portable Avatar.

Table 2 gives the list of PSD AIMs with their input and output Data.

AIW	AIMs	Name and Specification	JSON
PAF-PSD		Personal Status Display	<u>X</u>
	MMC-PDX	Personal Status Demultiplexing	<u>X</u>
	MMC-TTS	Text-to-Speech	<u>X</u>
	PAF-EFD	Entity Face Description	<u>X</u>
	PAF-EBD	Entity Body Description	<u>X</u>
	PAF-PMX	Portable Avatar Multiplexing	<u>X</u>

Table 2 – AIMs of Personal Status Display Composite AIM and JSON Metadata

7.12.5 JSON Metadata

https://schemas.mpai.community/PAF/V1.4/AIMs/PersonalStatusDisplay.json

7.12.6 Profiles

The Profiles of Personal Status Display are specified.

7.12.7 Conformance Testing

The Conformance Testing Method for the PAF-PSD Basic AIM is provided here. The Conformance Testing Method for the individual Basic AIMs of the PAF-PSD Composite AIM is provided by the individual Basic AIMs.

Table 2 provides the Conformance Testing Method for PAF-PSD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for PAF-PSD AIM

Receives Machine ID	Shall be string or validate against Instance ID Schema
Text Object	Shall validate against Text Object Schema. Text Data shall conform with Speech Qualifier.
Personal Status	Shall validate against Personal Status Schema.
Avatar Model	Shall validate against 3D Model Schema. Avatar Model Data shall conform with 3D Model Qualifier.
Speech <u>Model</u>	Shall validate against Machine Learning Model Schema. Speech Model Data shall conform with Machine Learning Model Qualifier.
Produces Portable Avatar	Shall validate against Portable Avatar Schema. Portable Avatar Data shall conform with respective Qualifiers.

7.13 Service Participant Authentication

7.13.1 Functions

Service Participant Authentication (PAF-SPA):

Receives	Participant ID	ID of Service Participant.
Uses	Face and Speech Objects	to find the correspondence between the Participant ID of a Participant requesting access to Service and the Subscriber ID of the Service Provider.
Produces	Subscriber ID	The Subscriber identified by the PAS-SPA AIM.

7.13.2 Reference Model

Figure 1 specifies the Reference Model of Service Participant Authentication (PAF-SPA).



Figure 1 - The Service Participant Authentication (PAF-SPA) AIM

7.13.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Service Participant Authentication (PAF-SPA) AIM.

Table 1 - Service Participant Authentication (PAF-SPA) AIM

Input	Description
Participant <u>ID</u>	ID of a Participant in a session of a Service. From an upstream AIM or another AIW.
Speech Object	Speech segment of Participant.
Face Visual Object	Face of Participant.
Output	Description
Subscriber ID	ID of Service Subscriber

7.13.4 JSON Metadata

 $\underline{https://schemas.mpai.community/PAF/V1.4/AIMs/ServiceParticipantAuthentication.json}$

7.13.5 Conformance Testing

Table 2 provides the Conformance Testing Method for PAF-SPA AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 - Conformance Testing Method for PAF-SPA AIM

Receives Participant ID	Shall be string or validate against Instance ID Schema.
Speech Object	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.
Face Visual Object	Shall validate against Visual Object Schema. Speech Data shall conform with Visual Qualifier.
Produces Subscriber ID	Shall be string.

7.13.6 Performance Assessment

Performance Assessment of a PAF-SPA AIM Implementation shall be performed using the subsets of two datasets of:

- 1. Speech segments all in the same language for each segment of which the Identity of the Speaker is provided with reference to a Taxonomy.
- 2. Faces for each face of which the Identity of the face is provided with reference to a Taxonomy

for which the identity of a speech segment's speaker is univocally connected to the identity of a face and vice-versa.

The Performance Assessment Report of a PAF-SPA AIM Implementation shall include:

- 1. The Identifier of the PAF-SPA AIM.
- 2. The identifier of the speech segment dataset.
- 3. The identifier of the face dataset.
- 4. The Taxonomy of speaker identifiers.
- 5. The Taxonomy of face identifiers.
- 6. The Accuracy of the Service Participant Authentication AIM defined as the number of correctly recognised pairs of speech segments and faces of the same Speaker and Face to the total number of submitted Speech and Face pairs.

7.14 Visual Scene Creation

7.14.1 Functions

Visual Scene Creation (PAF-VSC) V1.4 adds Avatars with specified Spatial Attitudes to a Visual Scene with given Visual Descriptors and produces Visual Scene Descriptors with the Spatial Attitudes of the Avatars' mouths:

Receives Avatar ID	Avatar Identifier
Audio-Visual Scene Descriptors	Descriptors of the full AV Scene.
Avatars	Avatars in the Visual Scene

	Spatial Attitude	Spatial Attitudes of Avatars
Produces	Visual Scene Descriptors	where the Avatars have been added to input AV Scene Descriptors
	Spatial Attitude of Avatars' Mouths	The Mouths of the Avatars' Spatial Attitude

7.14.2 Reference Model

Figure 1depicts the Reference Architecture of the Visual Scene Creation AIM ((PAF-VSC)).



Figure 1- The Visual Scene Creation (PAF-VSC) AIM

7.14.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Visual Scene Creation (PAF-VSC) AIM.

Table 1– I/O Data of the Visual Scene Creation (PAF-VSC) AIM

Input	Description
Avatar IDs	Identifiers of Avatars being handled.
Audio-Visual Scene Descriptors	Descriptors of the Initial Visual Scene
Avatars	Avatars of Participants and VMS
Spatial Attitudes	Spatial Attitudes of Participants' and VMS' Avatars
Output	Description
Visual Scene Descriptors	Descriptors of the entire Visual Scene
Mouth Spatial Attitudes	The Spatial Attitudes of the Mouths of Participants' and VMS' Avatars

7.14.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.4/AIMs/VisualSceneCreation.json

7.14.5 Conformance Testing

Table 2 provides the Conformance Testing Method for PAF-VSC AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 - Conformance Testing Method for PAF-VSC AIM

Receives	Avatar ID	Shall be string or validate against Instance ID Schema.
	Audio Visual Scene Descriptors	Shall validate against AV Scene Descriptors Schema.
	<u>Avatar</u> s	Shall validate against Avatar Schema. Avatar Data shall conform with 3D Model Qualifier.
	Face Descriptors	Shall validate against Face Descriptors Schema.
Produces	Visual Scene Descriptors	Shall validate against Visual Scene Descriptors Schema.
	<u>Spatial Attitude</u> of Avatars' Mouths	Shall validate against Spatial Attitude Schema.

8 Data Types

8.1 Avatar

8.1.1 Definition

A Data Type that includes:

- 1. ID of the Virtual Space in which the Avatar id located,
- 2. ID of the Avatar
- 3. Space-Time information of the Avatar
- 4. 3D Model of the Avatar
- 5. Body Descriptors of the Avatar
- 6. Face Descriptors of the Avatar.

8.1.2 Functional Requirements

Avatar conveys the following information:

- 1. The ID of the Virtual Space.
- 2. The ID of the Avatar.
- 3. The Time and Spatial Attitude of the Avatar is in the M-Instance.
- 4. The set of Data characterising an Avatar:
 - 1. 3D Model.
 - 2. Face Descriptors.
 - 3. Body Descriptors.

An Avatar Model of a human may:

- 1. Faithfully reproduce the visual appearance of the human.
- 2. Have their visual appearance altered, compared to that of the human.
- 3. Have an unrelated visual appearance.
- 4. Display a presumptive Personal Status in speech, face, and gesture.

8.1.3 Syntax

https://schemas.mpai.community/PAF/V1.4/data/Avatar.json

8.1.4 Semantics

Label	Size	Description
Header	N1 Bytes	Avatar Header.
- Standard-Avatar	9 Bytes	The characters "PAF-AVT-V"
– Version	N2 Bytes	Major version
– Dot-separator	1 Byte	The character "."
- Subversion	N3 Byte	Minor version
MInstanceID	N4 Bytes	ID of Virtual Space the Avatar belongs to.
AvatarSpaceTime	N5 Bytes	The inherent Space-Time info of the Avatar.
AvatarID	N6 Bytes	Identifier of Avatar.
AvatarData	N7 Bytes	Set of Data related to Avatar
- AvatarModel	N8 Bytes	Model of Avatar.
- BodyDescriptors	N9 Bytes	Avatar Body Descriptors.
- FaceDescriptors	N10 Bytes	Avatar Face Descriptors of Avatar.
DescrMetadata	N11 Bytes	Descriptive Metadata

8.1.5 Conformance Testing

A Data instance Conforms with Avatar (PAF-AVT) V1.4 if:

- 1. JSON Data validate against the Avatar's JSON Schema.
- 2. All Data in the Avatar's JSON Schema
 - 1. Have the specified type.
 - 2. Validate against their JSON Schemas.

8.2 Body Descriptors

8.2.1 Definition

Body Descriptors is a Data Type digitally representing a human or a humanoid.

Gesture Descriptors is a Data Type representing the subset of Body Descriptors selected by an application to convey Gesture information.

8.2.2 Functional Requirements

Body Descriptors should enable the representation of the joints of a body.

8.2.3 Syntax

Syntax is given by <u>Reference</u>. The Body Descriptors XML Syntax is given by: <u>https://www.web3d.org/x3d/content/examples/X3dResources.html</u>

8.2.4 Semantics

The semantics of Body Descriptors is provided by <u>https://www.web3d.org/content/hanim-architecture-v2</u>.

8.2.5 Conformance Testing

A Data instance Conforms with Body Descriptors (PAF-BDD) V1.4 if the Data instance validates against the Body Descriptors XML Schema.

8.3 Face Descriptors

8.3.1 Definition

Face Descriptors is a Data Type representing the features of the Face of an Entity.

8.3.2 Functional Requirements

The Face Descriptors represent the effect of the motion of the muscles of a human face.

The Face Descriptors Syntax represents the Actions Units of the Facial Action Coding System (FACS) originally developed by Carl-Herman Hjortsjö, adopted by Paul Ekman and Wallace V. Friesen (1978) and updated by <u>Ekman, Friesen, and Joseph C. Hager</u> (2002).

8.3.3 Syntax

https://schemas.mpai.community/PAF/V1.4/data/FaceDescriptors.json

8.3.4 Semantics

Hea	ader	N1 Bytes	Orientation FaceDescriptors
- St Fac	andard- eDescriptors	9 Bytes	The characters "OSD-FCD-V"
- Ve	ersion	N2 Bytes	Major version – 1 or 2 characters
- Do	ot-separator	1 Byte	The character "."
- Su	lbversion	N3 Bytes	Minor version – 1 or 2 characters
Fac	eDescriptorsID	N4 Bytes	Identifier of Face Descriptors.
AU	Description	N5 Bytes	Facial muscle generating the Action
1	Inner Brow Raiser	1 Byte	Frontalis, pars medialis
2	Outer Brow Raiser	1 Byte	Frontalis, pars lateralis
4	Brow Lowerer	1 Byte	Corrugator supercilii, Depressor supercilii
5	Upper Lid Raiser	1 Byte	Levator palpebrae superioris
6	Cheek Raiser	1 Byte	Orbicularis oculi, pars orbitalis

7	Lid Tightener	1 Byte	Orbicularis oculi, pars palpebralis
9	Nose Wrinkler	1 Byte	Levator labii superioris alaquae nasi
10	Upper Lip Raiser	1 Byte	Levator labii superioris
11	Nasolabial Deepener	1 Byte	Zygomaticus minor
12	Lip Corner Puller	1 Byte	Zygomaticus major
13	Cheek Puffer	1 Byte	Levator anguli oris (a.k.a. Caninus)
14	Dimpler	1 Byte	Buccinator
15	Lip Corner Depressor	1 Byte	Depressor anguli oris (a.k.a. Triangularis)
16	Lower Lip Depressor	1 Byte	Depressor labii inferioris
17	Chin Raiser	1 Byte	Mentalis
18	Lip Puckerer	1 Byte	Incisivii labii superioris and Incisivii labii inferioris
20	Lip stretcher	1 Byte	Risorius with platysma
22	Lip Funneler	1 Byte	Orbicularis oris
23	Lip Tightener	1 Byte	Orbicularis oris
24	Lip Pressor	1 Byte	Orbicularis oris
25	Lips part	1 Byte	Depressor labii inferioris or relaxation of Mentalis, or Orbicularis oris
26	Jaw Drop	1 Byte	Masseter, relaxed Temporalis and internal Pterygoid
27	Mouth Stretch	1 Byte	Pterygoids, Digastric
28	Lip Suck	1 Byte	Orbicularis oris
41	Lid droop	1 Byte	Relaxation of Levator palpebrae superioris
42	Slit	1 Byte	Orbicularis oculi
43	Eyes Closed	1 Byte	Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis
44	Squint	1 Byte	Orbicularis oculi, pars palpebralis
45	Blink	1 Byte	Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis
46	Wink	1 Byte	Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis
61	Eyes turn left	1 Byte	Lateral rectus, medial rectus
62	Eyes turn right	1 Byte	Lateral rectus, medial rectus
63	Eyes up	1 Byte	Superior rectus, Inferior oblique
64	Eyes down	1 Byte	Inferior rectus, Superior oblique

8.3.5 Conformance Testing

A Data instance Conforms with Face Descriptors (PAF-FCD) V1.4 if:

- The Data validates against the Face Descriptors' JSON Schema.
 All Data in the Face Descriptors' JSON Schema
 - - 1. Have the specified type.
 - 2. Validate against their JSON Schemas.

8.3.6 Mapping of AUs to Personal Status (Informative)

MPAI has defined a set of Cognitive States, Emotions, and Social Attitudes included in <u>Personal</u> <u>Status</u>. The Table below offers an informative mapping of some elements of Personal Status to Action Units (from <u>1</u>).

Personal Status	Cognitive State	Emotion	Prototypical (and variant AUs)
Нарру		17	12, 25 [6 (51%)]
Sad		32	4, 15 [1 (60%), 6 (50%), 11 (26%), 17 (67%)]
Fearful		13	1, 4, 20, 25 [2 (57%), 5 (63%), 26 (33%)]
Angry		2	4, 7, 24 [10 (26%), 17 (52%), 23 (29%)]
Surprised	18		1, 2, 25, 26 [5 (66%)]
Disgusted		11	9, 10, 17 [4 (31%), 24 (26%)]

This Table was obtained through a series of experiments with human subjects. AUs used by a subset of the subjects are shown in brackets with the percentage of the subjects using this less common AU in parentheses.

[1] Compound facial expressions of emotion | PNAS

8.4 Gesture Descriptors

8.4.1 Definition

Body Descriptors is a Data Type digitally representing a human or a humanoid.

Gesture Descriptors is a Data Type representing the subset of Body Descriptors selected by an application to convey Gesture information.

8.4.2 Functional Requirements

Body Descriptors should enable the representation of the joints of a body.

8.4.3 Syntax

Syntax is given by <u>Reference</u>. The Body Descriptors XML Syntax is given by: <u>https://www.web3d.org/x3d/content/examples/X3dResources.html</u>

8.4.4 Semantics

The semantics of Body Descriptors is provided by <u>https://www.web3d.org/content/hanim-architecture-v2</u>.

8.4.5 Conformance Testing

A Data instance Conforms with Body Descriptors (PAF-BDD) V1.4 if the Data instance validates against the Body Descriptors XML Schema.

8.5 MoCap Object

8.5.1 Definition

MoCap Object is a Data Type digitally representing the movement of a human.

8.5.2 Functional Requirements

MoCap Object represents the motion of the joints of a body.

8.5.3 Syntax

https://schemas.mpai.community/PAF/V1.4/MoCapObject.json

8.5.4 Semantics

Label	Size	Description
Header	N1 Bytes	The Header of the MoCap Object
- Standard-MoCapObject	9 Bytes	The characters "PAF-MCP-V"
– Version	N2 Bytes	Major version
- Dot-separator	1 Byte	The character "."
– Subversion	N3 Byte	Minor version
MInstanceID	N4 Bytes	The ID of the M-Instance.
MoCapObjectID	N5 Bytes	Identifier of the MoCap Object.
MoCapQualifier	N6 Bytes	Set of Data related to MoCap.
DescrMetadata	N7 Bytes	Descriptive Metadata

8.6 Portable Avatar

8.6.1 Definition

A Data Type that includes:

- 1. A set of avatar-related Data: M-Instance ID, Avatar ID, Space-Time, Avatar, Language Selector, Text, Speech Object, Personal Status, and
- 2. Descriptors of the Audio-Visual Scene where the Avatar is embedded and its Space-Time information.

8.6.2 Functional Requirements

Portable Avatar provides the following set of Data characterising a speaking avatar in a virtual space (M-Instance):

1. The ID of the virtual space (M-Instance) where the Portable Avatar is to be placed.

- 2. The space and time information of the "environment" to be placed in the M-Instance.
- 3. The Audio-Visual Scene representing the "environment".

4. The space and time information of the Avatar in the scene.

5. The Avatar represented as a 3D Model, its Face Descriptors and Body Descriptors.

6. The Language Preference of the Avatar.

7. The Text Object the Avatar is associated with, or which will be converted into a Speech Object.

8. The Speech Model used to synthesise the Text Object.

9. The Speech Object alternative to the Text Object that the Avatar utters.

10. The Personal Status of the Avatar.

8.6.3 Syntax

https://schemas.mpai.community/PAF/V1.4/data/PortableAvatar.json

8.6.4 Semantics

Label	Size	Description
Header	N1 Bytes	The Header of the Portable Avatar Data.
– Standard-PortableAvatar	9 Bytes	The characters "PAF-PAV-V"
- Version	N2 Bytes	Major version
– Dot-separator	1 Byte	The character "."
- Subversion	N3 Byte	Minor version
MInstanceID	N4 Bytes	The ID of the M-Instance.
PortableAvatarID	N5 Bytes	Identifier of the Portable Avatar.
PortableAvatarData	N6 Bytes	Set of Data related to Avatar.
- AudioVisualSceneSpaceTime	N7 Bytes	Space and Time info of AV Scene in M-Instance.
- AudioVisualSceneDescriptors	N8 Bytes	AV Scene Descriptors.
- AvatarSpaceTime	N9 Bytes	Space-Time of Avatar instance in AV Scene.
- Avatar	N10 Bytes	Avatar's Model and Face and Body Descriptors.
- LanguageSelector	N11 Bytes	Avatar's Language Preference.
- TextObject	N12 Bytes	Text associated with Avatar.
- SpeechObject	N13 Bytes	Set of Data related to Speech Object.
- SpeechModel	N14 Bytes	Neural Network Model for Speech Synthesis.
- PersonalStatus	N15 Bytes	Personal Status of Avatar.
DescrMetadata	N16 Bytes	Descriptive Metadata

8.6.5 Conformance Testing

A Data instance Conforms with Portable Avatar (PAF-PAV) V1.4 if:

1. JSON Data validate against the Portable Avatar 's JSON Schema.

- 2. All Data in the Portable Avatar 's JSON Schema

 - Have the specified type.
 Validate against their JSON Schemas.
 Conform with their Data Qualifiers if present.