Moving Picture, Audio and Data Coding
by Artificial Intelligence
www.mpai.community

MPAI Technical Specification

Human and Machine Communication (MPAI-HMC)

V2.1

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 Notices and Disclaimers.

Technical Specification:
Human and Machine Communication (MPAI-HMC)
V2.1

Contents

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. Adhere to a rigorous standard development process.
3. Be friendly 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. Be attractive to different industries, end users, and regulators.
5. Address 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. Provide 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 MPAI Patent Policy specifies the MPAI standard development process and the Framework Licence development guidelines.
2. *Technical Specification: Artificial Intelligence Framework (MPAI-AIF)* V2.1 specifies an environment enabling initialisation, dynamic configuration, and control of AI applications 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.

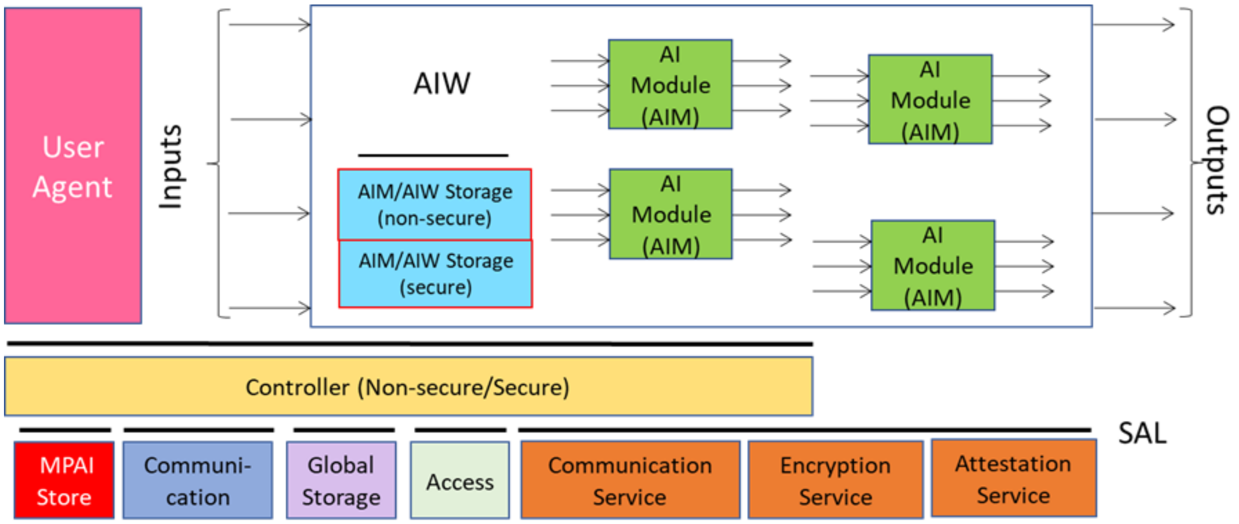

Figure 1 – The AI Framework (MPAI-AIF) V2 Reference Model

3. [*Technical Specification: Data Types, Formats, and Attributes (MPAI-TFA) V1.4*]() 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. [*Technical Specification: Governance of the MPAI Ecosystem (MPAI-GME) V2.0*]() defines the following elements:
   1. Standards, i.e., the ensemble of Technical Specifications, Reference Software, Conformance Testing, and Performance Assessment.
   2. Developers of MPAI-specified AIMs and Integrators of MPAI-specified AIWS (Implementers).
   3. MPAI Store in charge of making AIMs and AIWs submitted by Implementers available to Integrators and End Users.
   4. Performance Assessors, 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.

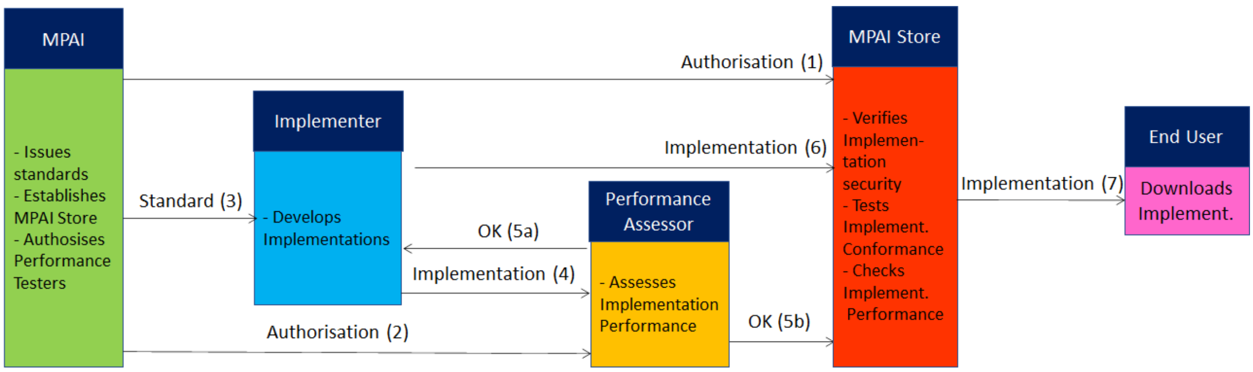

Figure 2 – The MPAI Ecosystem

2 Introduction (informative)

Technical Specification: Human and Machine Communication (MPAI-HMC) V2.1 specifies two AI Modules and reuses AI Modules and Data Types from other MPAI Technical Specifications to enable advanced forms of communication between humans and machines and applies those AI Modules and Data Types to the Communicating Entities in Context (HMC-CEC) Use Case. The referenced Technical Specifications are:

1. Two foundational MPAI Technical Specifications:
    1. AI Framework (MPAI-AIF) V2.1.
    2. Data Types, Formats, and Qualifiers (MPAI-TFA) V1.2
2. Four application MPAI Technical Specifications:
    1. Context-based Audio Enhancement (MPAI-CAE) V2.4.
    2. Multimodal Conversation (MPAI-MMC) V2.4.
    3. Object and Scene Description (MPAI-OSD) V1.4.
    4. Portable Avatar Format (MPAI-PAF) V1.5.
3. One ancillary MPAI Technical Specification AI Module Profiles (MPAI-PRF) V1.0.

Capitalised Terms are defined in Table 1 if they are MPAI-HMC-specific and online for all MPAI-defined Terms. Non-capitalised terms have the commonly intended meaning.

This Chapter is Informative. The Chapters and Sections of this Technical Specification are Normative unless they are explicitly identified as Informative.

3 Scope

Technical Specification: Human and Machine Communication (MPAI-HMC) V2.1 – referred to in the following as MPAI-HMC V2.1 or MPAI-HMC – specifies technologies that enable advanced forms of communication between humans in a real space or represented in a Virtual Space, and Machines represented as humanoids in a Virtual Space or rendered as humanoids in a real space.

The HMC communicating parties are generically called *Entities*. To the extent possible, MPAI-HMC strives to be neutral to the nature - real or virtual - of an Entity.

The word *Communication* is to be intended to mean that an Entity should have the capability to understand the semantics of the media involved - Text, Speech, Audio, Visual, and 3D Models. Different MPAI-HMC implementations may provide different levels of understanding and responding. At this state, MPAI-HMC does not provide the means to assess an Entity's capability to understand and provide pertinent responses.

MPAI-HMC V2.1 applies the technologies from HMC (AI Modules) and other MPAI Technical Specifications (AI Modules and Data Types) to the *Communicating Entities in Context* Use Case as a first instance offering effective forms of human-Machine communication that can be implemented in an interoperable fashion.

MPAI-HMC has been developed by CAE-DC, MMC-DC, PAF-DC, and the MPAI-OSD group of the Requirements Standing Committee.

MPAI may develop future versions of this Technical Specification or new Technical Specifications related to the Human and Machine Communication area.

4 Definitions

Capitalised terms are defined in Table 1 if they are MPAI-HMC-specific and in Definitions which defines the terms used in all current MPAI Technical Specifications. Non-capitalised terms have the commonly intended meaning.

Table 1 - MPAI-HMC specific terms

Term	Definition
Communication Item	An element generated by a Machine communicating with an Entity expressed by a Portable Avatar.
Context	Information describing the attributes and the environment of an Entity, such as language, culture etc.
Culture	The collection of ideas, language, customs, and social behaviour governing the way a human, or a group of humans perceive, express and behave.

5 References

This page provides normative and information references. The full set of non-MPAI normative references can be accessed online.

5.1 Normative References

1. MPAI; Technical Specification: Governance of the MPAI Ecosystem (MPAI-GME) V1.1.
2. MPAI; Technical Specification: Artificial Intelligence Framework (MPAI-AIF) V2.2.
3. MPAI; Technical Specification: Context-based Audio Enhancement (MPAI-CAE) V2.4.
4. MPAI; Technical Specification: Multimodal Conversation (MPAI-MMC) V2.4.
5. MPAI; Technical Specification: Object and Scene Description (MPAI-OSD) V1.4.
6. MPAI; Technical Specification: Portable Avatar Format (MPAI-PAF) V1.5.
7. MPAI; Technical Specification: Profiles (MPAI-PRF) - AI Modules (PRF-AIM) V1.1.
8. MPAI; Technical Specification: Data Types, Formats, and Attributes (MPAI-TFA) V1.4.

5.2 Informative References

9. MPAI; The MPAI Statutes.
10. MPAI; The MPAI Patent Policy.
11. MPAI; Technical Specification: MPAI Metaverse Model (MPAI-MMM) – Technologies V2.1.
12. MPAI; Technical Specification: Neural Network Watermarking (MPAI-NNT) - Neural Network Traceability (NNW-NNT) V1.1.
13. MPAI; Technical Specification: Neural Network Traceability (MPAI-NNW) V1.0.

6 AI Workflows

6.1 Technical Specification

Technical Specification: Multimodal Conversation (MPAI-HMC) V2.1 assumes that workflows are based on *Technical Specification: AI Framework (MPAI-AIF) V2.1* specifying the standard AI Framework (AIF) that enables initialisation, dynamic configuration, execution, and control of AI Workflows (AIW) composed of interconnected AI Modules (AIM).

Table 1 provides the link to the currently specified AI Workflow. The provides the AIW function, reference model, Input/Output Data, Functions of AIMs used, Input/Output Data of AIMs, and links to the AIW-related AIMs, and JSON metadata.

All AI-Workflows specified by MPAI-HMC V2.1 supersede those specified by previous MPAI-MMC specifications which can still be used if their version is explicitly indicated.

Table 1 - AIWs of MPAI-HMC V2.1

Acronym	Title	JSON
HMC-CEC	Communicating Entities in Context	X

6.1.1 Communicating Entities in Context

6.1.1.1 Functions

The *Communicating Entities in Context* (HMC-CEC) AI Workflow enables Entities to communicate with other Entities, possibly in different Contexts, where:

1. **Entity** refers to one of:
    1. human
        1. In an *audio-visual scene*, or
        2. Represented as a Digitised Human in an *Audio-Visual Scene*.
    2. Digital Human representing
        1. A human as a Digitised Human in an Audio-Visual Scene, or
        2. A Machine as a Virtual Human in an Audio-Visual Scene.
    3. A Machine not represented.
2. **Context** is information describing the attributes of an Entity, such as language, culture etc.

Note that the same non-capitalised and capitalised word represents an object in the **real world** and its digital representation in the **Virtual World**, respectively.

Depending on its real or virtual nature, an Entity communicates with another Entity by:

1. Using the human's body, speech, context, and the audio-visual scene the human is immersed in
2. Rendering the Virtual Entity as a speaking humanoid in an audio-visual scene, or
3. Communicating by emitting Communication Items.

Communication Item is an implementations of Portable Avatar, a Data Type including Data related to an Avatar and its Context, that enables a receiver to render an Avatar as intended by the sender.

HMC-CEC assumes that:

1. *Input/Output Audio* and *Input/Output Visual* are Audio Object or Speech Objects and Visual Object, respectively.
2. The *real space*
    1. Is digitally represented as an Audio-Visual Scene that includes the communicating human.
    2. May include other humans and generic objects.
3. The *Virtual Space*
    1. Contains a Digital Human and/or its Speech components in an Audio-Visual Scene.
    2. May include other Digital Humans and generic Objects.
4. The *Machine* can:
    - Understand the semantics of the Communication Item at different layers of depth depending on the technologies used by an Implementation.
    - Produce a multimodal response expected to be congruent with the received information.
    - Render the response as a speaking Virtual Human in an Audio-Visual Scene.
    - Convert the Data produced by an Entity to Data whose semantics is compatible with the Context of another Entity.

Note 1: An AI Module is specified only by its Functions and Interfaces. Implementers are free to use their preferred technologies to achieve the expected AIM Functions while respecting the Function and Interface constraints.
Note 2: An Implementation may subdivide a given AIM into more than one AIM, provided that their combined Functions and Interface conform with the Interfaces of the corresponding HMC-CEC AIM.
Note 3: An implementation may combine AIMs into one, provided that the resulting AIM performs the combine Functions and exposes the Interface of the combined HMC-CEC AIMs.

6.1.1.2 Reference Model

Figure 1 depicts the Reference Model of the *Communicating Entities in Context (HMC-CEC)* AIW that includes *AI Modules (AIM)* per *Technical Specification: AI Framework (MPAI-AIF) V2.2*. Three out of the seven AIMs in Figure 1 (*Audio-Visual Scene Description*, *Entity Context Understanding*, and *Personal Status Display*) are Composite AIMs, i.e., they include interconnected AIMs.

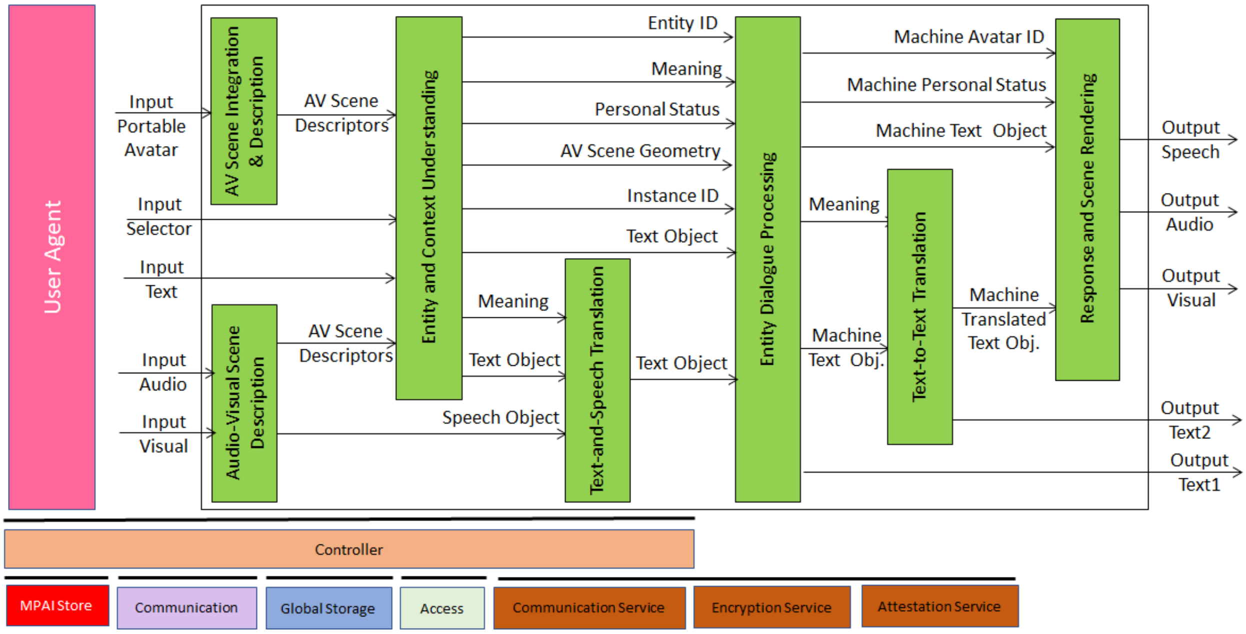

Figure 1 – Communicating Entities in Context (HMC-CEC) AIW

Note that:

1. The *Input Selector* enables an Entity to inform the Machine about use of media types (Text. Speech, and Visual), Portable Avatar, and Language Preferences.
2. The Machine captures the information emitted by an Entity and its Context through *Input Text*, *Input Audio* and *Input Visual*. In Figure 1 Audio includes Speech.
3. The *Input Portable Avatar* is the Communication Item emitted by a Machine Entity.
4. The *Audio-Visual Scene Descriptors* are digital representations of a real audio-visual scene or a Virtual Audio-Visual Scene produced either by the *Audio-Visual Scene Description* AIM or the *Audio-Visual Scene Integration and Description* AIM.
5. To facilitate identification, AIMs are labelled with three letters indicating the Technical Specification that specifies it, followed by a hyphen "-", followed by three letters uniquely identifying the AIM defined by that Technical Specification. For instance, Portable Avatar Demultiplexing is indicated as PAF-PDX where PAF refers to *Technical Specification: Portable Avatar Format (MPAI-PAF)* and PDX refers to the [Portable Avatar Demultiplexing]() AIM also specified by MPAI-PAF.

6.1.1.3 Input/Output Data

Table 1 gives the Input/Output Data of the MPAI-HMC AIW.

Table 1 – Input/Output Data of the HMC-CEC AIW

Input	**Description**
Portable Avatar	A Communication Item emitted by the Entity communicating with the ego Entity.
Input Selector	Selector containing data specifying the media and the language used in the communication.
Input Text	Text Object generated by the communicating Entity as information additional to or in lieu of Speech Object.
Input Audio	The audio scene captured by the Machine.
Input Visual	The visual scene captured by the Machine.
Output	**Description**
Portable Avatar	The Communication Item produced by the Machine.
Output Speech	The speech corresponding to the Speech Object in the output Communication Item.
Output Audio	The audio corresponding to the Audio Object in the output Communication Item.
Output Visual	The visual corresponding to the Visual Object in the output Communication Item.
Output Text1	The Text contained in a Communication Item or associated with Output Audio and Output Visual.
Output Text2	Translation of Output Text2.

6.1.1.4 Functions of AI Modules

Table 2 gives the functions of HMC-CEC AIMs.

Table 2 – Functions of AI Modules

AIM	**Functions**
Audio-Visual Scene Integration and Description	Produces Audio-Visual Scene Descriptors where the Avatar in Portable Avatar has been added to Audio-Visual Scene.
Audio-Visual Scene Description	Provides Audio-Visual Scene Descriptors.
Entity and Context Understanding	Provides information on Entity and its Context.
Text-and-Speech Translation	Produces translation of Entity Speech or Text.
Entity Dialogue Processing	Produces Text and Personal Status of Machine in response to input from Entity and Context Understanding.
Text-to-Text Translation	Produces translation of Machine Text using Text and Meaning.
Personal Status Display	Produces Portable Avatar.

Audio-Visual Scene Rendering	Renders the content of the internally generated Portable Avatar.

6.1.1.5 Input/Output Data of AI Modules

Table 3 gives the I/O Data of the AIMs of HMC-CEC. Note that an ID can either be specified as an Instance Identifier or refer to a generic identifier.

Table 3 - Input/Output Data of AI Modules

AIM	Receives	Produces
Audio-Visual Scene Integration and Description	Input Portable Avatar	Audio-Visual Scene Descriptors
Audio-Visual Scene Description	Input Audio Input Visual	Audio-Visual Scene Descriptors
Entity and Context Understanding	Audio-Visual Scene Descriptors Input Text Input Selector	Audio-Visual Scene Geometry Personal Status Entity ID Text Object Meaning Instance Identifier
Text-and-Speech Translation	Meaning Text Object Speech Object	Text Object
Entity Dialogue Processing	Audio-Visual Scene Geometry Personal Status Entity ID Text Meaning Instance Identifier	Machine Personal Status Machine Avatar ID Machine Text Object Output Text
Text-to-Text Translation	Machine Text Object Machine Personal Status	Machine Translated Text Object
Personal Status Display	Machine Personal Status Machine Avatar ID Machine Text Object	Output Portable Avatar
Audio-Visual Scene Rendering	Output Portable Avatar	Output Audio Output Visual

6.1.1.6 AIW, AIMs, and JSON Metadata

Table 4 provides the list of AIMs composing the HMC-CEC AIW. The AIMs of a Composite AIM are also provided down to the level of Basic AIMs.

Table 4 - AIW, AIMs, and JSON Metadata

AIW	AIMs/1	AIMs/2	AIMs/3	Name	JSON
HMC-CEC				Communicating Entities in Context	X

	HMC-SID			AV Scene Integration and Description	X
	OSD-AVS			Audio-Visual Scene Description	X
		CAE-ASD		Audio Scene Description	X
			CAE-AAT	Audio Analysis Transform	X
			CAE-ASL	Audio Source Localisation	X
			CAE-ASE	Audio Separation and Enhancement	X
			CAE-AST	Audio Synthesis Transform	X
			CAE-AMX	Audio Descriptors Multiplexing	X
		OSD-VSD		Visual Scene Description	X
		OSD-AVA		Audio-Visual Alignment	X
	HMC-ECU			Entity And Context Understanding	X
		OSD-SDX		Audio-Visual Scene Demultiplexing	X
		MMC-ASR		Automatic Speech Recognition	X
		OSD-VOI		Visual Object Identification	X
			OSD-VDI	Visual Direction Identification	X
			OSD-VOE	Visual Object Extraction	X
			OSD-VII	Visual Instance Identification	X
		CAE-AOI		Audio Object Identification	X
		MMC-NLU		Natural Language Understanding	X
		MMC-PSE		Personal Status Extraction	X
			MMC-ETD	Entity Text Description	X
			MMC-ESD	Entity Speech Description	X
			PAF-EFD	Entity Face Description	X
			PAF-EBD	Entity Body Description	X
			MMC-PTI	PS-Text Interpretation	X
			MMC-PSI	PS-Speech Interpretation	X
			PAF-PFI	PS-Face Interpretation	X
			PAF-PGI	PS-Gesture Interpretation	X
			MMC-PMX	Personal Status Multiplexing	X

		MMC-TTT		Text-to-Text Translation	X
	MMC-TST			Text-and-Speech Translation	
		MMCASR		Automatic Speech Recognition	X
		MMC-TTT		Text-to-Text Translation	X
		MMC-ESP		Entity Speech Description	X
		MMC-TSD		Text-to-Speech with Descriptors	X
	MMC-EDP			Entity Dialogue Processing	X
	MMC-TTT			Text-to-Text Translation	X
	PAF-RSR			Response and Scene Rendering	X
	PAF-PSD			Personal Status Display	X
		MMC-TTS		Text-to-Speech	X
		PAF-IFD		Entity Face Description	X
		PAF-IBD		Entity Body Description	X
		PAF-PMX		Portable Avatar Multiplexing	X

6.1.1.7 Reference Software

6.1.1.8 Conformance Testing

Table 5 provides the Conformance Testing Method for HMC-CEC 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 5 – Conformance Testing Method for HMC-CEC AIM

Receives	Portable Avatar	Shall validate against Portable Avatar Schema. Portable Avatar Data shall conform with respective Qualifiers.
	Input Selector	Shall validate against Selector schema.
	Input Text	Shall validate against Text Object schema. Audio Data shall conform with Text Qualifier.
	Input Audio	Shall validate against Audio Object schema. Audio Data shall conform with Audio Qualifier.
	Input Visual	Shall validate against Visual Object schema. Audio Data shall conform with Visual Qualifier.
Produces	Portable Avatar	Shall validate against Portable Avatar Schema. Portable Avatar Data shall conform with respective Qualifiers.
	Output Audio	Shall validate against Audio Object schema. Audio Data shall conform with Audio Qualifier.

	Output Visual	Shall validate against Visual Object schema. Audio Data shall conform with Visual Qualifier.
	Output Text	Shall validate against Text Object schema. Audio Data shall conform with Text Qualifier.

6.2 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 that those of the MPAI-HMC Standard and does not guarantee that it is secure.
3. Users shall verify that they have the right to use any third-party software required by the Reference Software Implementation.
4. Users should note that the Reference Software Implementation may require the acceptance of licences from third-party repositories.

Note that, at this stage, the MPAI-HMC AIW is only partly implemented.

6.3 Conformance Testing

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

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.

6.4 Performance Assessment

Performance is a multidimensional entity because it can have various connotations. Therefore, the Performance Assessment Specification provides 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 Communicating Entities in Context AIW can measure how well the AIW responds to a question.
2. Bias: Performance of a Communicating Entities in Context AIW can measure the quality of responses in dependence of the type of message received.
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, a limited number of MPAI-HMC AIMs include a Performance Assessment Specification.

7 AI Modules

7.1 Technical Specifications

Table 1 provides the links to the specifications and the JSON syntax of all AIMs specified by ***Technical Specification: Human and Machine Communication (MPAI-HMC) V2.1***. The MPAI-HMC V2.1 AI-Modules supersede those specified by earlier versions than V2.1. They may still be used if their version is explicitly signalled. The Composite AIMs are in **bold**.

Table 1 - Basic and Composite AI Modules

Acronym	Specification	JSON
HMC-ECU	**Entity and Context Understanding**	**X**
HMC-SID	AV Scene Integration and Description	X

Table 2 provides the full list of with web links to the AI Modules utilised by HMC-CEC organised according to the Technical Specifications specifying them.

Table 2 - AI Modules utilised by HMC-CEC organised by Technical Specifications

CAE	MMC	OSD	PAF
Audio Analysis Transform	Automatic Speech Recognition	Audio Scene Description	Audio-Visual Scene Rendering
Audio Descriptors Multiplexing	Entity Dialogue Processing	Audio-Visual Alignment	Face Identity Recognition
Audio Object Identification	Entity Speech Description	Audio-Visual Event Description	Entity Body Description
Audio Separation and Enhancement	Entity Text Description	Audio-Visual Scene Demultiplexing	Entity Face Description
Audio Source Localisation	Natural Language Understanding	**Audio-Visual Scene Description**	Portable Avatar Demultiplexing
Audio Synthesis Transform	**Personal Status Extraction**	Speech Scene Description	PS-Face Interpretation
HMC	Personal Status Multiplexing	Visual Direction Identification	PS-Gesture Interpretation
AV Scene Integration and Description	PS-Speech Interpretation	Visual Instance Identification	**Personal Status Display**
Entity and Context Understanding	PS-Text Interpretation	Visual Object Extraction	Portable Avatar Multiplexing
	Speaker Identity Recognition	**Visual Object Identification**	
	Text and Speech Translation	Visual Scene Description	
	Text-To-Speech		
	Text-to-Text Translation		

7.1.1 Entity and Context Understanding

7.1.1.1 Function

Entity and Context Understanding (HMC-ECU) enables a Machine

1. To understand the information conveyed by an Entity and its Context, in the form of either:
    1. An Audio-Visual Scene (if Entity is a human).
    2. A Portable Avatar (if Entity is a machine).
2. To produce a pertinent response composed of Machine Text and Machine Personal Status.

Therefore, Entity and Context Understanding (HMC-ECC):

Receives	Audio-Visual Scene Descriptors	And separates into components.
Recognises	Speech	Of Entity.
	Audio Object and Visual Object.	Providing their Identities.
Understands	Natural Language	(Of Entity) expressed as Text being cognizant of the Audio and Visual Instances
Extracts	Personal Status.	Of Entity.
Translates	Text	Of Entity.
Produces:	Audio-Visual Scene Geometry	Geometry of the Scene.
	Entity ID	Entity producing Input Data.
	Audio Instance ID	Identified Instance.
	Visual Instance ID	Identified Instance.
	Personal Status	On Entity.
	Translated Text	Of Refined Text.
	Meaning	Of Refined Text.

7.1.1.2 Reference Model

Entity and Context Understanding (HMC-ECU) is an AIM whose Reference Model is depicted in Figure 2.

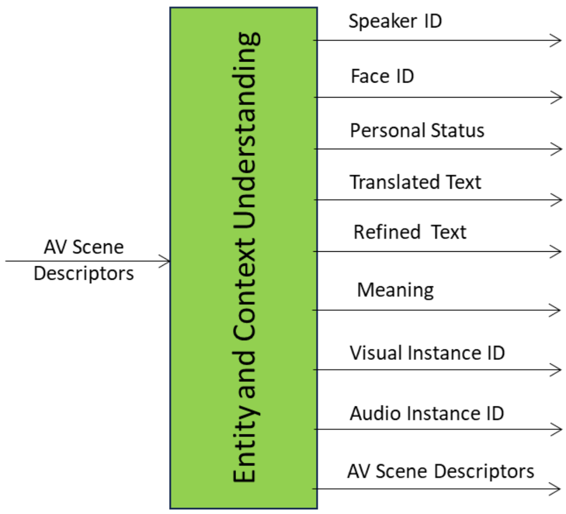

Figure 1 – The Entity and Context Understanding (HMC-ECU) AIM

7.1.1.3 Input and Output Data

Table 1 specifies the Input and Output Data of the of the Entity Context Understanding (HMC-ECU) AIM.

Table 1 – I/O Data of the Entity Context Understanding (HMC-ECU) AIM

Input	Description
Body DescriptorsObject	The Descriptors of the Body Objects of Entities in the Visual Scene.
Face DescriptorsObject	The Descriptors of the Face Objects of Entities in the Visual Scene.
Speech Object	The digital representation of the speech emitted by the Entity.
Audio-Visual Scene Geometry	The digital representation of the spatial arrangement of the Audio, Visual, and Audio-Visual Objects of the Scene.
Visual Objects	The Visual Objects of the Scene.
Audio Object	The Audio Objects of the Scene.
Text Object	Text of Entity with Entity ID.
Output	**Description**
Personal Status	Personal Status of Entity having the Entity ID.
Translated Text Object	Translated Text of Text Object or of Text conveyed by Speech Object.
Refined Text Object	Refined Text of Speech Object.
Meaning	Other name for Refined Text Descriptors.
Visual Instance Identifier	The Identifier of the specific Visual Object belonging to a level in the taxonomy.
Audio-Visual Scene Geometry	As in Input
Audio Instance Identifier	The Identifier of the specific Audio Object belonging to a level in the taxonomy.

7.1.1.4 SubAIMs

Entity and Context Understanding (HMC-ECU) is a Composite AIM whose Reference Model is depicted in Figure 2.

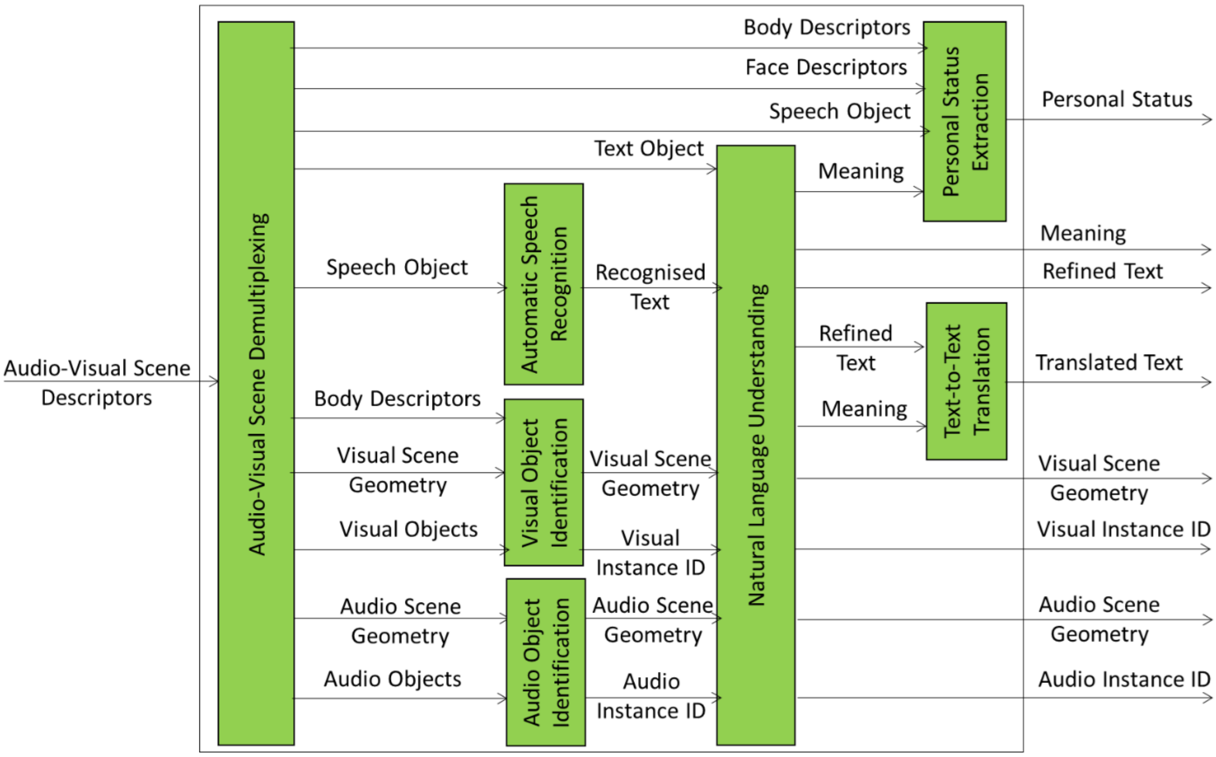

Figure 2 – The Entity and Context Understanding Composite (HMC-ECU) AIM

Table 2 - AIMs and JSON Metadata

AIM	AIM/1	AIM/2	AIM Name	JSON
HMC-ECU			Entity and Context Understanding	X
	OSD-SDX		Audio-Visual Scene Demultiplexing	X
	MMC-ASR		Automatic Speech Recognition	X
	OSD-VOI		Visual Object Identification	X
	CAE-AOI		Audio Object Identification	X
	MMC-NLU		Natural Language Understanding	X
	MMC-PSE		Personal Status Extraction	X
		MMC-ETD	Entity Text Description	X
		MMC-ESD	Entity Speech Description	X
		PAF-EFD	Entity Face Description	X
		PAF-EBD	Entity Body Description	X
		MMC-PTI	PS-Text Interpretation	X
		MMC-PSI	PS-Speech Interpretation	X
		PAF-PFI	PS-Face Interpretation	X
		PAF-PGI	PS-Gesture Interpretation	X
		MMC-PMX	Personal Status Multiplexing	X
	MMC-TTT		Text-to-Text Translation	X

7.1.1.5 JSON Metadata

https://schemas.mpai.community/HMC/V2.1/AIMs/EntityAndContextUnderstanding.json

7.1.1.6 Profiles

Entity Context Understanding Profiles are defined.

7.1.1.7 Conformance Testing

Table 2 provides the Conformance Testing Method for the HMC-ECU 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 CAE-ECU AIM

Receives	Body Descriptors	Shall validate against Body Descriptors XML Schema.
	Face Descriptors	Shall validate against Face Descriptors Schema.
	Speech Object	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.
	Audio-Visual Scene Geometry	Shall validate against Audio-Visual Scene Geometry Schema.
	Visual Objects	Shall validate against Visual Object Schema. Visual Data shall conform with Visual Qualifier.
	Audio Object	Shall validate against Audio Object Schema. Audio Data shall conform with Visual Qualifier.
	Text Object	Shall validate against Text Object Schema. Text Data shall conform with Visual Qualifier.
Produces	Personal Status	Shall validate against Personal Status Schema.
	Translated Text	Shall validate against Text Object Schema. Text Data shall conform with Visual Qualifier.
	Refined Text	Shall validate against Text Object Schema. Text Data shall conform with Visual Qualifier.
	Meaning	Shall validate against Meaning schema
	Visual Instance ID	Shall validate against Instance ID schema.
	Audio-Visual Scene Geometry	Shall validate against Audio-Visual Scene Geometry Schema.
	Audio Instance ID	Shall validate against Instance ID schema.

7.1.1.8 Performance Assessment

7.1.2 AV Scene Integration and Description

7.1.2.1 Functions

The AV Scene Integration and Description (HMC-SID) AIM performs the following functions:

Receives	Portable Avatar
Adds	The Avatar in the Input Portable Avatar to the Audio-Visual Scene Descriptors conveyed by the Input Portable Avatar with an appropriate Spatial Attitude. If the Input Portable Avatar does not include a Scene, the AV Scene Integration and Description AIM uses a generic Scene.

Produces	The Audio-Visual Scene Descriptors of the resulting Audio-Visual Scene.

7.1.2.2 Reference Model

Figure 1 depicts the HMC-SID Reference Architecture.

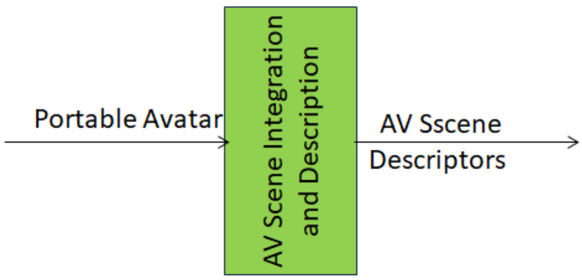

Figure 1 – The AV Scene Integration and Description AIM

7.1.2.3 I/O Data

The Input and Output Data of are specified in Table 1.

Table 1 – I/O Data of the AV Scene Integration and Description AIM

Input	Description
Portable Avatar	A Communication Item received from a Machine Entity.
Output	**Description**
Audio-Visual Scene Descriptors	The Descriptors of the AV Scene where the Avatar conveyed by the Input Portable Avatar has been added to the Scene with the appropriate Spatial Attitude.

7.1.2.4 JSON Metadata

https://schemas.mpai.community/HMC/V2.2/AIMs/AVSceneIntegrationAndDescription.json

7.1.2.5 Conformance Testing

Table 2 provides the Conformance Testing Method for HMC-SID 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 HMC-SID AIM

Receives	Portable Avatar	Shall validate against Portable Avatar Schema. Portable Avatar Data shall conform with respective Qualifiers.

Produces	[Audio-Visual Scene Descriptors]()	Shall validate against AV Scene Descriptors Schema.

7.1.2.6 Performance Assessment

7.1.3 Audio Analysis Transform

7.1.3.1 Function

Audio Analysis Transform (CAE-AAT) receives n Audio Object (Multichannel Audio), into frequency bands via a Fast Fourier Transform (FFT), and produces an Audio Object In the Transform domain.

Receives	Audio Object	*As Multichannel Audio*
Transforms	Multichannel Audio	into frequency bands via a Fast Fourier Transform (FFT). The following operations are carried out in discrete frequency bands. When such a configuration is used, a 50% overlap between subsequent audio blocks needs to be employed. The output is a data structure comprising complex valued audio samples in the frequency domain.
Produces	Audio Object	In the Transform domain.

7.1.3.2 Reference Model

Figure 1 depicts the Reference Architecture of the Audio Analysis Transform (CAE-AAT) AIM.

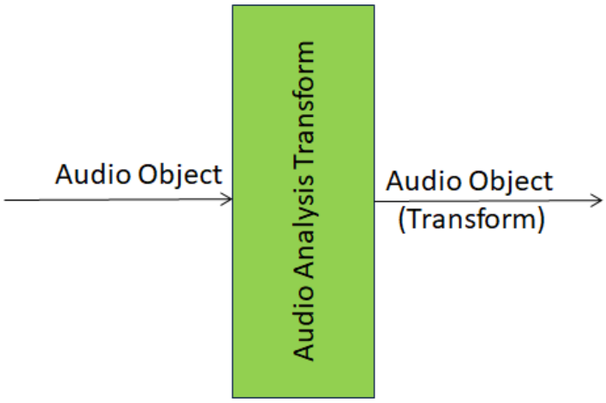

Figure 1 – Audio Analysis Transform (CAE-AAT) AIM

7.1.3.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio Analysis Transform (CAE-AAT) AIM.

Table 1 – Audio Analysis Transform (CAE-AAT) AIM

Input	**Description**
[Audio Object]()	Audio Object (with associated Microphone Array info)
Output	**Description**

Audio Object (Transform)	The result of the application of the Fast Fourier Transform to Multichannel Audio.

7.1.3.4 JSON Metadata

https://schemas.mpai.community/CAE1/V2.4/AIMs/AudioAnalysisTransform.json

7.1.3.5 Reference Software

The Audio Analysis Transform Reference Software can be downloaded from the MPAI Git.

7.1.3.6 Conformance Testing

Table 2 – Conformance Testing Method for CAE-AAT AIM

Receives	Audio Object (Microphone Array)	Shall validate against Audio Object schema. The Format of the Audio Data shall conform with the Format specified by Audio Qualifier.
Produces	Audio Object (Transform)	Shall validate against Audio Object schema. The Format of the Audio Data shall conform with the Format specified by Audio Qualifier.

7.1.3.7 Performance Assessment

The following steps shall be followed when assessing the Performance of a CAE-AAT AIM instance.

1. Use the following datasets:
    1. DS1: *n* Test Audio Object files including Multichannel Audio as Interleaved Multichannel Audio format.
    2. DS2: *n* Expected Audio Object Output files including data in Transform Interleaved Multichannel Audio format.
2. Feed the AIM under test with the Test files (DS1).
3. Perform the following steps to analyse the Audio Object (Transform) with the Expected Audio Objects (DS2):
    1. Check the data format of the Audio Object (Transform) with the format of the given Expected Audio Objects.
    2. Calculate the peak-to-peak Amplitude (A) of each Audio block in the Expected Audio Objects.
    3. Calculate the RMSE of each Audio block by comparing the Audio Object (Transform) (x) with the Expected Audio Objects (y).
    4. Accept the AIM under test if, for each audio block, these two conditions are satisfied:
        1. Data format of the Audio Object (Transform) is the same as the format of the Expected Audio Object and
        2. RMSE < A* 0.1%.
4. The Performance Assessor will provide the following matrix containing a limited number of input records (n) with the corresponding outputs. If an input record fails, the tester would specify the reason why the test case fails.

Input data (DS1)	**Expected Output Data (DS2)**	**Data Format**	**RMSE**

Audio Object (Microphone Array) ID_1	Audio Object (Transform) ID_1	T/F	< *A*0.1%*
Audio Object (Microphone Array) ID_2	Audio Object (Transform) ID_2	T/F	< *A*0.1%*
Audio Object (Microphone Array) ID_3	Audio Object (Transform) ID_3	T/F	< *A*0.1%*
…	…	…	…
Audio Object (Microphone Array) ID_n	Audio Object (Transform) ID_n	T/F	< *A*0.1%*

5. Final evaluation: T/F Denoting with *i*, the record number in DS1 and DS2, the matrices reflect the results obtained with input records i with the corresponding outputs i.

DS1	DS2	Audio Object (Transform) output value (from AIM under test)
DS1[*i*]	DS2[*i*]	Audio Object (Transform)[*i*]

Table 2 provides the Performance Assessment Method for the formats of the CAE-AAT AIM output.
Note: If a schema contains references to other schemas, performance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and perform with the Qualifier, if present.

7.1.4 Audio Descriptors Multiplexing

7.1.4.1 Functions

Audio Descriptor Multiplexing (CAE-AMX) multiplexes Enhanced Audio Objects and their Geometry into out Audio Scene Descriptors:

Receives	*Enhanced Audio Objects*	Audio Objects with reduced noise.
	Audio Scene Geometry	The spatial arrangement of Audio Objects.
Multiplexes	*Enhanced Audio Objects*	Enhanced-quality Audio Objects
	Audio Scene Geometry	Arrangement of Audio Objects
Produces	*Audio Scene Descriptors*	The Descriptors of the Audio Scene.

7.1.4.2 Reference Model

Figure 1 depicts the Reference Architecture of the Audio Descriptor Multiplexing AIM.

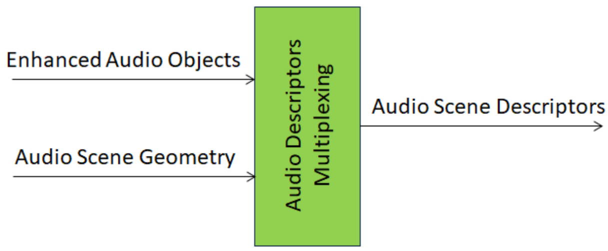

Figure 1 – The Audio Descriptor Multiplexing AIM

7.1.4.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio Descriptor Multiplexing AIM.

Table 1 – I/O Data of Audio Descriptor Multiplexing

Input	**Description**
Enhanced Audio Object	Time-domain Audio Objects without noise.
Audio Scene Geometry	The Space-Time arrangement of Audio objects in an Audio Scene
Output	**Description**
Audio Scene Descriptors	The combination of Audio Scene Geometry and Audio Objects.

7.1.4.4 JSON Metadata

https://schemas.mpai.community/CAE1/V2.4/AIMs/AudioDescriptorsMultiplexing.json

7.1.4.5 Conformance Testing

Table 2 – Conformance Testing Method for CAE-AMX AIM

Receives	Enhanced Audio Objects	Shall validate against Audio Object schema. Audio Data shall conform with Audio Qualifier.
	Audio Scene Geometry	Shall validate against Audio Basic Scene Geometry schema.
Produces	Audio Scene Descriptors	Shall validate against Audio Basic Scene Descriptors schema.

7.1.4.6 Performance Assessment

The following steps shall be followed when assessing the Performance of a CAE-AMX AIM instance.

1. Use the following datasets:
    1. DS1: *n* Enhanced Audio Objects Test files.
    2. DS3: *n* Audio Scene Geometry of the Enhanced Audio Objects.
    3. DS4: *n* Expected Output Audio Scene Descriptors.
2. Feed the AIM under test with the Test files (DS1, DS3).
3. Analyse the Audio Scene Descriptors with the Expected Audio Scene Descriptors (DS4).
    1. Check the Audio Scene Descriptors with Expected given Audio Scene Descriptors.
    2. Calculate the peak-to-peak Amplitude (A) of each Audio block in the Expected Audio Scene Descriptors.
    3. Calculate the RMSE of each Audio block by comparing the output (x) with the Audio Scene Descriptors (y) Audio blocks.
    4. Accept the CAE-AMX AIM under test if, for each audio block, these the two conditions are satisfied:
        1. Data format of Audio Scene Descriptors is the same as the Expected Audio Scene Descriptors and
        2. RMSE < A * 0.1%
4. The Conformance Tester will provide the following matrix with a limited number of input records (n) with the corresponding outputs. If an input record fails, the tester would specify the reason why the test case fails.

Input data (DS1, DS3)	Expected Output Data (DS4)	Data Format	RMSE
Enhanced Audio Objects ID_1 Audio Scene Geometry ID_1	Audio Scene Descriptors ID_1	T/F	< A * 0.1%
Enhanced Audio Objects ID_2 Audio Scene Geometry ID_2	Audio Scene Descriptors ID_2	T/F	< A * 0.1%
Enhanced Audio Objects ID_3 Audio Scene Geometry ID_3	Audio Scene Descriptors ID_3	T/F	< A * 0.1%
…	…		…
Enhanced Audio Objects ID_n Audio Scene Geometry ID_n	Audio Scene Descriptors ID_n	T/F	< A * 0.1%

5. Final evaluation: T/F Denoting with *i*, the record number in DS1 and DS3, the matrices reflect the results obtained with input records i with the corresponding outputs i.
6.

DS1	DS3	DS4	CAE-AMX output value (obtained through the AIM under test)
DS1[*i*]	DS3[i]	DS4[i]	Multiplexer[*i*]

Table 3 provides the Conformance Testing Method for the formats of the CAE-AMX AIM output.

Note: 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.

7.1.5 Audio Object Identification

7.1.5.1 Functions

Audio Object Identification (CAE-AOI) receives Audio Objects and their Scene Geometry and produces the Identities of the Audio Objects:

Receives	*Audio Scene Geometry*	The spatial arrangements of the Audio Objects.
	Audio Objects	The individual input Audio Objects
Identifies	The Audio Objects.	Provides Audio Object Identifiers
Produces	The *Audio Instance IDs*	The Instance Identifier of the Audio Objects.

7.1.5.2 Reference Model

Figure 1 depicts the Reference Architecture of the Audio Object Identification AIM.

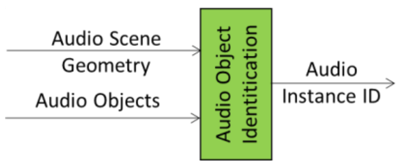

Figure 1 – Audio Object Identification AIM

The Audio Object Identification AIM shall be able to parse either an Audio-Visual Scene Geometry or its Audio Scene Geometry subset.

7.1.5.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio Object Identification AIM.

Table 1 – I/O Data of the Audio Object Identification AIM

Input	Description
Audio Scene Geometry	The digital representation of the spatial arrangement of the Visual Objects of the Scene.
Audio Objects	The Audio Objects in the Audio Scene Geometry with an identifiable source target of identification.
Output	**Description**
Audio Instance Identifier	The Instance Identifier of the specific Audio Object.

7.1.5.4 JSON Metadata

https://schemas.mpai.community/CAE1/V2.4/AIMs/AudioObjectIdentification.json

7.1.5.5 Conformance Testing

Table 2 provides the Conformance Testing Method for the CAE-AOI AIM. Conformance Testing of the individual AIMs of the CAE-AOI 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 CAE-AOI AIM

Receives	Audio Scene Geometry	Shall validate against Audio Basic Scene Geometry schema.
	Audio Objects	Shall validate against Audio Object schema. Audio Data shall conform with Audio Qualifier.
Produces	Audio Instance Identifier	Shall validate against Instance ID schema.

7.1.5.6 Performance Assessment

7.1.6 Audio Separation and Enhancement

7.1.6.1 Functions

Audio Separation and Enhancement (CAE-ASE) receives Audio Objects in the Transform domain with their Spatial Attitude and produces Enhanced Audio Objects with their Scene Geometry.

Receives	*Audio Objects*	in the Transform domain.
	Audio Spatial Attitudes	Spatial Attitudes of the input Audio Objects.
Separates	*Audio Objects*	by using their Spatial Attitudes.
Produces	*Enhanced Audio Object*	in the Transform domain.
	Audio Scene Geometry	The Geometry of Audio Objects in the Scene.

7.1.6.2 Reference Model

Figure 1 depicts the Reference Architecture of the Audio Separation and Enhancement AIM.

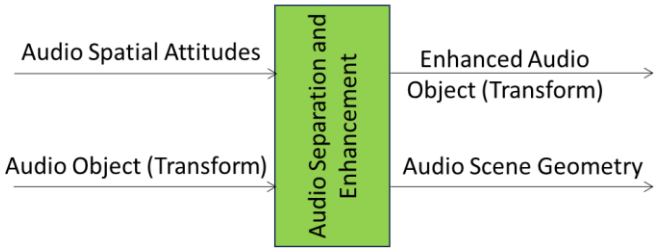

Figure 1 – Audio Separation and Enhancement AIM

7.1.6.3 Input/Output Data

Table 11 specifies the Input and Output Data of the Audio Separation and Enhancement AIM.

Table 1 – I/O Data of Audio Separation and Enhancement

Input	Description
Audio Object	The result of the application of the Fast Fourier Transform to the Multichannel Audio.
Audio Spatial Attitudes	The Spatial Attitudes of Audio Objects.
Output	**Description**
Enhanced Audio Object	Enhanced Multichannel Audio in the transform domain.
Audio Scene Geometry	The spatial arrangement of the Audio Objects.

7.1.6.4 SubAIMs

No SubAIMs.

7.1.6.5 JSON Metadata

https://schemas.mpai.community/CAE/V2.4/AIMs/AudioSeparationAndEnhancement.json

7.1.6.6 Conformance Testing

Table 2 provides the Conformance Testing Method for the formats of the CAE-ASE AIM output.
Note: 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 CAE-ASE AIM

Receives	Audio Object	Shall validate against Audio Object schema. Audio Data shall conform with Audio Qualifier.
	Audio Spatial Attitudes	Shall validate against Spatial Attitude schema.
Produces	Enhanced Audio Object	Shall validate against Audio Object schema. Audio Data shall conform with Audio Qualifier.
	Audio Scene Geometry	Shall validate against Audio Basic Scene Geometry schema.

7.1.6.7 Performance Assessment

The following steps shall be followed when assessing the Performance of a CAE-ASE AIM instance.

1. Use the following datasets:
    1. DS1: *n* Test files containing Audio Objects (Transform).
    2. DS4: *n* Test files containing the Spatial Attitudes of Audio Objects.
    3. DS2: *n* Expected Enhanced Audio Objects (Transform) Files.
    4. DS3: *n* Expected Audio Scene Geometries.
2. Feed the AIM under test with the Test Audio Objects (Transform) and Spatial Attitudes.
3. Analyse the Audio Scene Geometry and Enhanced Audio (Transform).
    1. Control the Audio Scene Geometry with the Expected Audio Scene Geometry:
        1. Count the number of Audio Objects in the Audio Scene Geometry.
        2. Calculate the angle difference (AD) in degrees between the Audio Objects (u) in the Audio Scene Geometry and the Audio Objects (v) in the Expected Audio Scene Geometry.
    2. Compare the number of Audio Blocks in the Expected Audio Objects with the number of Audio Blocks in the Audio Objects (Transform).
    3. Calculate Signal to Interference Ratio (SIR), Signal to Distortion Ratio (SDR), and Signal to Artefacts Ratio (SAR) between the Expected Audio Objects (Transform) and Output Audio Objects (Transform).
    4. Accept the CAE-ASE AIM under test if these four conditions are satisfied:
        1. The number of Audio Objects (Transform) in the Audio Scene Geometry is equal to the number of Audio Objects (Transform) in the Expected Audio Scene Geometry.
        2. The number of Audio Blocks in the Audio Objects (Transform) is equal to the number of Audio Blocks in the Expected Audio Objects (Transform).
        3. Compare each Audio Objects (Transform) in the Audio Scene Geometry with the Audio Objects (Transform) in the Expected Audio Scene Geometry.
            1. Each Audio Objects (Transform)'s AD between the Expected and Output is less than 5 degrees.
        4. Compare each Audio Objects (Transform) with the Audio Objects (Transform) in the Expected Audio Objects (Transform).
            1. If the room reverb time (T60) is greater than 0.5 seconds.
                1. Each object's SIR between the Expected and Output is greater than or equal to 10 dB.
                2. Each object's SDR between the Expected and Output is greater than or equal to 3 dB.
                3. Each object's SAR between the Expected and Output is greater than or equal to 3 dB.
            2. If the room reverb time (T60) is less than 0.5 seconds.

   1. Each object’s SIR between the Expected and Output is greater than or equal to 15 dB.
   2. Each object’s SDR between the Expected and Output is greater than or equal to 6 dB.
4. The Performance Assessor shall provide the following matrix containing a limited number of input records (*n*) with the corresponding outputs. If an input record fails, the tester would specify the reason why the test case fails.

Input data (DS1, DS4)	**Expected Output Data (DS2, DS3)**	**Data Format**	**Audio Scene Geometry**	**Source Separation Metrics**
Spatial Attitude (ID_1) Audio Object (Transform) ID_1	Enhanced Audio Object (Transform) ID_1 Audio Scene Geometry ID_1	T/F	T/F	T/F
Spatial Attitude (ID_2) Audio Object (Transform) ID_2	Enhanced Audio Object (Transform) ID_2 Audio Scene Geometry ID_2	T/F	T/F	T/F
Spatial Attitude (ID_3) Audio Object (Transform) ID_3	Enhanced Audio Object (Transform ID_3 Audio Scene Geometry ID_3	T/F	T/F	T/F
…	…	…	…	…
Spatial Attitude (ID_n) Audio Object (Transform) ID_n	Enhanced Audio Object (Transform ID_n Audio Scene Geometry ID_n	T/F	T/F	T/F

6. Final evaluation : T/F Denoting with *i*, the record number in DS1, DS2, and DS3, the matrices reflect the results obtained with input records i with the corresponding outputs i.

DS1	DS2	DS3	Sound Field Description output value (obtained through the AIM under test)
DS1[*i*]	DS2[*i*]	DS3[i]	SpeechDetectionandSeparation[*i*]

7.1.7 Audio Source Localisation

7.1.7.1 Functions

Audio Source Localisation (CAE-ASL) receives Audio Objects, detects the Audio Objects in the Audio Scene, and determines and produces as output their Spatial Attitudes:

Receives	*Audio Objects*	With associated Microphone Array information.
Detects	*Audio Objects*	In the Audio Scene.
Determines	*Spatial Attitudes*	Of Audio Objects.
Produces	*Spatial Attitudes*	Of input Audio Objects.

7.1.7.2 Reference Model

Figure 1 depicts the Reference Architecture of the Audio Source Localisation (CAE-ASL) AIM.

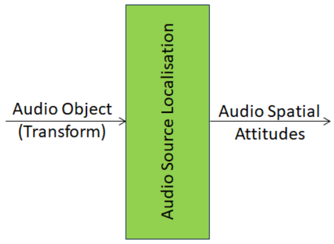

Figure 1 – Audio Source Localisation (CAE-ASL) AIM

7.1.7.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio Source Localisation (CAE-ASL) AIM.

Table 1 – Audio Source Localisation (CAE-ASL) AIM

Input	**Description**
Audio Object	The result of the application of the Fast Fourier Transform to the Multichannel Audio (with associated Microphone Array info).
Output	**Description**
Audio Spatial Attitudes	The Orientations and Directions of Audio Objects.

7.1.7.4 JSON Metadata

https://schemas.mpai.community/CAE1/V2.3/AIMs/AudioSourceLocalisation.json

7.1.7.5 Conformance Testing

The following procedure shall be followed when testing the Conformance of a CAE-ASL AIM instance.

1. Use the following datasets:
    1. DS1: *n* Test files containing Audio Objects (Transform) .
    2. DS2: *n* Expected Spatial Attitudes.
2. Feed the AIM under test with the Test files.
3. Analyse the Spatial Attitudes produced by the CAE-ASL AIM instance.
4. Calculate the angle difference (AD) in degrees between the output Spatial Attitudes with the Expected Spatial Attitudes.

Input data (DS1)	**Expected Output Data (DS2)**	**Data Format**	**RMSE**
Audio Object (Microphone Array) ID_1	Spatial Attitude ID_1	T/F	< *A*0.1%*
Audio Object (Microphone Array) ID_2	Spatial Attitude (Transform) ID_2	T/F	< *A*0.1%*
Audio Object (Microphone Array) ID_3	Spatial Attitude (Transform) ID_3	T/F	< *A*0.1%*
…	…	…	…
Audio Object (Microphone Array) ID_n	Spatial Attitude (Transform) ID_n	T/F	< *A*0.1%*

5. Final evaluation: T/F Denoting with *i*, the record number in DS1 and DS2, the matrices reflect the results obtained with input records i with the corresponding outputs i.

DS1	DS2	Audio Object (Transform) output value (from AIM under test)
DS1[*i*]	DS2[*i*]	Audio Object (Transform)[*i*]

Table 2 provides the Conformance Testing Method for the formats of the CAE-ASL AIM output. Note: 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 CAE-ASL AIM

Receives	Audio Object	Shall validate against Audio Object schema. Audio Data shall conform with Audio Qualifier.
Produces	Audio Spatial Attitudes	Shall validate against Spatial Attitude schema.

7.1.8 Audio Synthesis Trnsform

7.1.8.1 Functions

Audio Synthesis Transform (CAE-AST) receives an Enhanced Audio Object in the Transform domain, transforms the Audio Object back to the time domain and produces an Enhanced Audio Object with associated Microphone Array info:

Receives	*Enhanced Audio Object (Transform)*	with associated Microphone Array info.
Transforms	*Enhanced Audio Object (Transform)*	from the frequency domain to the time domain via an Inverse Fast Fourier Transform (IFFT).
Produces	*Enhanced Audio Object*	with associated Microphone Array info.

7.1.8.2 Reference Model

Figure 1 depicts the Reference Architecture of the Audio Synthesis Transform (CAE-AST) AIM.

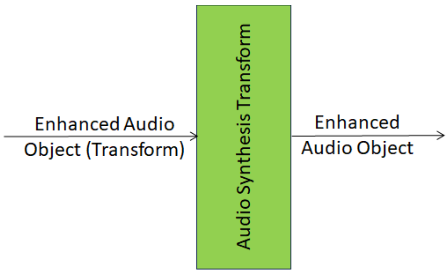

Figure 1 – The Audio Synthesis Transform (CAE-AST) AIM

7.1.8.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio Synthesis Transform (CAE-AST) AIM.

Table 1 – I/O Data of Synthesis Transform (CAE-AST) AIM

Input	Description
Enhanced Audio Objects (time-frequency)	Audio Objects in the time-frequency domain.
Output	**Description**
Enhanced Audio Objects (time)	Audio Objects in the time domain.

7.1.8.4 JSON Metadata

https://schemas.mpai.community/CAE1/V2.4/AIMs/AudioSynthesisTransform.json

7.1.8.5 Reference Software

The Audio Synthesis Transform Reference Software can be downloaded from the MPAI Git.

7.1.8.6 Conformance Testing

Receives	Enhanced Audio Objects (time-frequency)	Shall validate against Audio Object schema. The Format of the Audio Data shall conform with the Format specified by Audio Qualifier.
Produces	Enhanced Audio Objects (time)	Shall validate against Audio Object schema. The Format of the Audio Data shall conform with the Format specified by Audio Qualifier.

7.1.8.7 Performance Assessment

Table 61 gives the *Enhanced Audioconference Experience (CAE-EAE) Synthesis Transform* Means and how they are used.

Table 61 – AIM Means and use of Enhanced Audioconference Experience (CAE-EAE) Synthesis Transform

Means	Actions
Performance Testing Dataset	DS1: *n* Test files including data in Denoised Transform Speech format. DS2: *n* Expected Output files including data in Denoised Speech format.
Procedure	1. Feed the AIM under test with the Test files (DS1). 2. Analyse the Denoised Speech with the Expected Output files (DS2).
Evaluation	1. Check the Denoised Speech data format with the given Expected Output files format. 2. Calculate the peak-to-peak Amplitude (A) of each Audio block in the Expected Output files. 3. Calculate the RMSE of each Audio block by comparing the output (x) with the Expected Output files (y) Audio blocks.

	4. Accept the AIM under test if, for each audio block, these the two conditions are satisfied: a. Data format of the Denoised Speech is the same with the Expected Output Files and b. RMSE < A* 0.1%

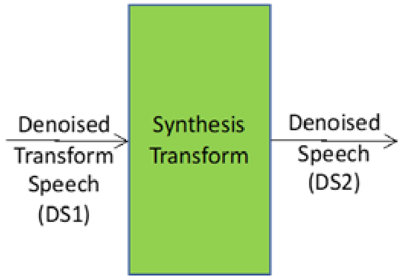

Figure 24 - Synthesis Transform Testing Flow

After the Tests, Performance Assessor shall fill out *Table 62Table 62*

Table 62 – Performance Assessment form of Enhanced Audioconference Experience (CAE-EAE) Synthesis Transform

Performance Assessor ID	Unique Performance Assessor Identifier assigned by MPAI
Standard, Use Case ID and Version	Standard ID and Use Case ID, Version and Profile of the standard in the form “CAE:EAE:1:0”.
Name of AIM	Synthesis Transform
Implementer ID	Unique Implementer Identifier assigned by MPAI Store.
AIM Implementation Version	Unique Implementation Identifier assigned by Implementer.
Neural Network Version*	Unique Neural Network Identifier assigned by Implementer.
Identifier of Performance Testing Dataset	Unique Dataset Identifier assigned by MPAI Store.
Test ID	Unique Test Identifier assigned by Performance Assessor.
Actual output	The Performance Assessor will provide the following matrix with a limited number of input records (*n*) with the corresponding outputs. If an input record fails, the tester would specify the reason why the test case fails. Input data (DS1) Expected Output Data (DS2) Data Format RMSE

	Denoised Transform Speech ID_1 / Denoised Speech ID_1 / T/F / < A* 0.1% Denoised Transform Speech ID_2 / Denoised Speech ID_2 / T/F / < A* 0.1% Denoised Transform Speech ID_3 / Denoised Speech ID_3 / T/F / < A* 0.1% … / … / / … Denoised Transfom Speech ID_n / Denoised Speech ID_n / T/F / < A* 0.1% Final evaluation: T/F Denoting with *i*, the record number in DS1 and DS2, the matrices reflect the results obtained with input records i with the corresponding outputs i. DS1 / DS2 / Synthesis Transform output value (obtained through the AIM under test) DS1[*i*] / DS2[*i*] / SynthesisTransform[*i*]
Execution time*	Duration of test execution.
Test comment*	Comments on test results and possible needed actions.
Test Date	yyyy/mm/dd.

* *Optional field*

7.1.9 Automatic Speech Recognition

7.1.9.1 Functions

The Automatic Speech Recognition (MMC-ASR) AIM extracts the text conveyed by an utterance (speech). The input speech may be accompanied by an auxiliary text, the identifier of the speaker the Speech Overlap data type and the time indicating the portion of the input speech that should be recognised:

Receives	Language Selector	Signalling the language of the speech.
	Auxiliary Text	Text that may be used to provide context information.
	Speech Object	Speech to be recognised.

	Speaker ID	ID of speaker uttering speech.
	Speech Overlap	Data type providing information of speech overlap.
	Speaker Time	Time during which the speech is to be recognised.
Produces	Recognised Text	(Also called text transcript).

Recognised Text can be a Text Segment or just a string.

7.1.9.2 Reference Model

Figure 1 depicts the Reference Model of the Automatic Speech Recognition (MMC-ASR) AIM.

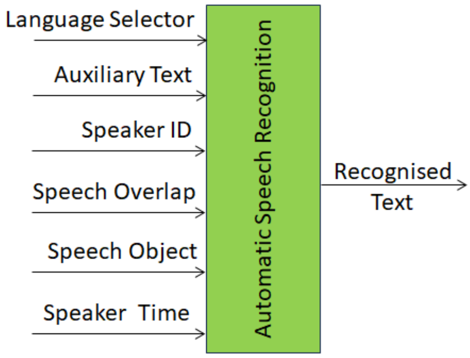

Figure 1 – The Automatic Speech Recognition (MMC-ASR) AIM

7.1.9.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Automatic Speech Recognition (MMC-ASR) AIM.

Table 1 – I/O Data of the Automatic Speech Recognition (MMC-ASR) AIM

Input	Description
Language Selector	Selects input language
Auxiliary Text Object	Text Object with content related to Speech Object.
Speech Object	Speech Object emitted by Entity
Speaker Identifier	Identity of Speaker
Speech Overlap	Times and IDs of overlapping speech segments
Speaker Time	Time during which Speech is recognised
Output	**Description**
Recognised Text Object	Output of the Automatic Speech Recognition AIM, a Text Segment or just a string.

7.1.9.4 JSON Metadata

https://schemas.mpai.community/MMC/V2.4/AIMs/AutomaticSpeechRecognition.json

7.1.9.5 Reference Software

7.1.9.5.1 Disclaimers

1. This MMM-ASR Reference Software Implementation is released with the BSD-3-Clause licence.
2. The purpose of this Reference Software is to demonstrate a working Implementation of MMC-ASR, 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.

7.1.9.5.2 Guide to the ASR code #1

The code takes Speech Objects from MMC-AUS and generates Text Segments (called text transcripts). It uses the whisper-large-v3 model to convert an input Speech Object (speaker's turn) into a Text Segment (here called text transcript). Disfluencies (e.g., repetitions, repairs, filled pauses) are often omitted. The Whisper reference document is available.
The MMC-ASR Reference Software is found at the MPAI gitlab site. Use of this AI Modules is for developers who are familiar with Python, Docker, RabbitMQ, and downloading models from HuggingFace. The Reference Software 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: commands for cloning https://huggingface.co/openai/whisper-large-v3

Library: https://github.com/linto-ai/whisper-timestamped

7.1.9.5.3 Guide to the ASR code #2

Use of this AI Modules is for developers who are familiar with Python and downloading models from HuggingFace,
A wrapper for the Whisper NN Module:
1. Manages input files and parameters: Speech Object
2. Performs Speech Recognition on each Speech Object by executing the Whisper Module.
3. Outputs Recognised Text.

The MMC-ASR Reference Software is found at the NNW gitlab site (registration required). It contains:
1. The python code implementing the AIM.
2. The required libraries are: pytorch and transformers (HuggingFace).

7.1.9.5.4 Acknowledgements

This version of the MMC-ASR Reference Software
1. #1 has been developed by the MPAI *AI Framework* Development Committee (AIF-DC).
2. #2 has been developed by the MPAI *Neural Network Watermarking* Development Committee (NNW-DC).

7.1.9.6 Conformance Testing

Table 2 provides the Conformance Testing Method for MMC-ASR 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 - MMC-ASR AIM Conformance Testing

Input	Language Selector	Shall validate against the Language Selector part of the schema.
	Auxiliary Text	Shall validate against the Text Object schema. Text Data shall conform with the Text Qualifier.
	Speech Object	Shall validate against the Speech Object schema. Speech Data shall conform with the Speech Qualifier.
	Speaker ID	Shall validate against the Instance ID schema.
	Speech Overlap	Shall validate against the Speech Overlap schema.
	Speaker Time	Shall validate against the Time schema.
Output	Text Object	Shall validate against the Text Object schema. Text Data shall conform with the Text Qualifier, e.g. output language shall be that indicated by the Language Selector,

Table 3 provides an example of MMC-ASR AIM Conformance Testing.

Table 3 - An example of MMC-ASR AIM Conformance Testing

Input Data	**Data Format**	**Input Conformance Testing Data**
Speech Object	.wav	All input Speech files to be drawn from Speech files.
Output Data	**Data Format**	**Output Conformance Testing Criteria**
Recognised Text	Unicode	All Text files produced shall conform with Text files.

7.1.9.7 Performance Assessment

Performance Assessment of an ASR Implementation (ASRI) can be performed for a language for which there is a dataset of speech segments of various durations with corresponding Transcription Text. An MMC-ASR AIM Performance Assessment Report shall be based on the following steps and specify the input dataset used.

For each Recognised Text produced by the ASRI being Assessed for Performance in response to a speech segment provided as input:

1. Compare the Recognised Text with the Transcription Text
2. Compute the Word Error Rate (WER) defined as the sum of deletion, insertion, and substitution errors in the Recognised Text compared to the Transcription Text, divided by the total number of words in the Transcription Text.

This code can be used to compute the WER.

Performance Assessment of an ASRI for a language in a Performance Assessment Report is defined as "The WER computed on all speech segments included in the reported dataset".

7.1.10 Entity Dialogue Processing

7.1.10.1 Functions

The Entity Dialogue Processing (MMC-EDP) AIM provides a text in response to an input text. The MMC-EDP AIM may also receive some or all of the following inputs: the descriptors (Meaning) of the input text, the ID of the speaker who produced the input text and the identifier of a face, a Personal Status, the Summary data type of the conversation being held in the scene, the Geometry of the objects of an audio-visual scene, the Identifiers of some of the objects in the scene. MMC-EDP may also produce the Personal Status of the MMC-EDP:

Receives	Text Object	Text of the entity upstream to be processed.
	Object Instance ID	Of an object in a scene.
	Personal Status	of the entity upstream.
	Text Descriptors	Descriptors of input Text Object.
	AV Scene Geometry	Geometry of the AV scene containing object whose ID is provided.
	Speaker ID	ID of speaker uttering the speech that contains the Text Object.
	Face ID	ID of the face of the speaker uttering the speech that contains the Text Object.
	Summary	A summary of the discussions being held in the environment.
Handles	One Text Object at a time	From an entity upstream.
Recognises	The identity	Of entity upstream using speech and/or face.
Takes into account	Past Text Objects	and their spatial arrangement.
Produces	Summary	Edited summary based on input data.
	Text Object	of Machine.
	Personal Status	of Machine.

7.1.10.2 Reference Model

Figure 1 depicts the Reference Model of the Entity Dialogue Processing (MMC-EDP) AIM.

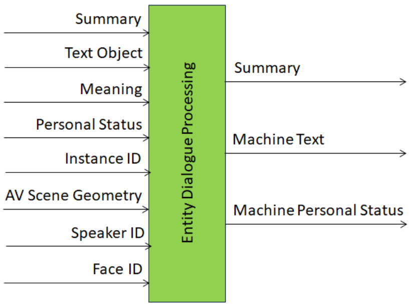

Figure 1 – Entity Dialogue Processing (MMC-EDP) AIM Reference Model

7.1.10.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Entity Dialogue Processing (MMC-EDP) AIM.

Table 1 – I/O Data of the Entity Dialogue Processing (MMC-EDP) AIM

Input	Description
Summary	The summary in the current state.
Text Object	Text or Refined Text from the Entity the Machine is communicating with.
Meaning	Descriptors of Text and/or Translated Text of the Entity the Machine is communicating with.
Personal Status	Personal Status of the Entity the Machine is communicating with.
Instance Identifier	ID of the Audio of Visual Object the Entity refers to.
Audio-Visual Scene Geometry	The Geometry of the AV Scene.
Speaker Identifier	The ID of the Speaker.
Face Identifier	The ID of the Face.
Output	**Description**
Machine Text Object	Text produced by the Machine in response to input.
Machine Personal Status	The Personal Status the Machine intends to add to its Modalities.
Summary	The result of refining the input Summary taking comments into consideration.

7.1.10.4 JSON Metadata

https://schemas.mpai.community/MMC/V2.4/AIMs/EntityDialogueProcessing.json

7.1.10.5 Profiles

Profiles of Entity Dialogue Processing are specified.

7.1.10.6 Conformance Testing

Table 2 provides the Conformance Testing Method for MMC-EDP 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 – MMC-EDP AIM Conformance Testing

Input	Text Object	Shall validate against Text Object schema. Text Data shall conform with Text Qualifier.
	Object Instance ID	Shall validate against Instance Identifier schema.
	Input Personal Status	Shall validate against Personal Status schema.
	Meaning	Shall validate against Text Descriptors schema.

	Audio-Visual Scene Geometry	Shall validate against AV Scene Geometry schema.
	Speaker ID	Shall validate against Instance ID schema.
	Face ID	Shall validate against Face ID schema.
	Summary	Shall validate against Summary schema. Shall validate against Text Object schema. Text Data shall conform with Text Qualifier.
Output	Edited Summary	Shall validate against Summary schema. Shall validate against Text Object schema. Text Data shall conform with Text Qualifier.
	Machine Text Object	Shall validate against Text Object schema. Text Data shall conform with Text Qualifier.
	Machine Personal Status	Shall validate against Personal Status schema.

Table 3 provides an example of MMC-EDP AIM Conformance Testing.

Table 3 – An example of MMC-EDP AIM Conformance Testing

Input Data	**Data Type**	**Input Conformance Testing Data**
Meaning	JSON	All input JSON Emotion files to be drawn from Meaning JSON Files
Recognised Text	Unicode	All input Text files to be drawn from Text files.
Input Emotion	JSON	All input JSON Emotion files to be drawn from Emotion JSON Files
Output Data	**Data Type**	**Output Conformance Testing Criteria**
Machine Text	Unicode	All Text files produced shall conform with Text.
Machine Emotion	JSON	Emotion JSON Files shall validate against Emotion Schema

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

7.1.11 Entity Speech Description

7.1.11.1 Functions

The Entity Speech Description (MMC-ESD) AIM receives an utterance (input speech) and produces the descriptors of the utterance:

Receives	*Speech Object*	From an AIM or Entity.
Produces	*Speech Descriptors*	of the input Speech Object.

7.1.11.2 Reference Model

Figure 1 depicts the Reference Model of the Entity Speech Description (MMC-ESD) AIM.

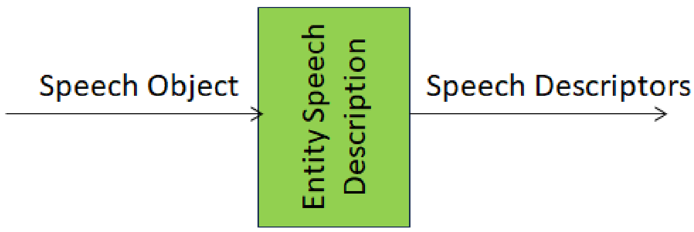

Figure 1 Entity Speech Description (MMC-ESD) AIM Reference Model

7.1.11.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Entity Speech Description (MMC-ESD) AIM.

Table 1 – I/O Data of the Entity Speech Description (MMC-ESD) AIM

Input	Description
Speech Object	Speech of Entity or AIM
Output	**Description**
Speech Descriptors	Descriptors of Entity Speech

7.1.11.4 JSON Metadata

https://schemas.mpai.community/MMC/V2.4/AIMs/EntitySpeechDescription.json

7.1.11.5 Conformance Testing

Table 2 provides the Conformance Testing Method for MMC-ESD 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-ESD AIM

Input	Speech Object	Shall validate against Speech Object schema. Speech Data shall conform with Speech Qualifier.
Output	Speech Descriptors	Shall validate against Speech Descriptors schema.

7.1.12 Entity Text Description

7.1.12.1 Functions

The Entity Text Description (MMC-ETD) AIM receives an input text and produces the descriptors of the input text:

Receives	*Text Object*	Text Object from an entity.
Produces	*Text Descriptors*	Descriptors of Text Object's Text Data.

7.1.12.2 Reference Model

Figure 1 depicts the Reference Model of the Entity Text Description (MMC-ETD) AIM.

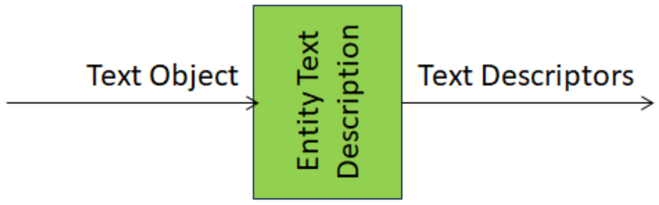

Figure 1 Entity Text Description (MMC-ETD) AIM Reference Model

7.1.12.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Entity Text Description (MMC-ETD) AIM.

Table 1 – I/O Data of the Entity Text Description (MMC-ETD) AIM

Input	Description
[Text Object](#)	Text Object from and entity.
Output	**Description**
[Text Descriptors](#)	Descriptors of Descriptors of Text Data of Text Object.

7.1.12.4 JSON Metadata

https://schemas.mpai.community/MMC/V2.4/AIMs/EntityTextDescription.json

7.1.12.5 Conformance Testing

Table 2 provides the Conformance Testing Method for MMC-ETD 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-ETD AIM

Input	[Text Object](#)	Shall validate against Text Object schema. Text Data shall conform with Text Qualifier.
Output	[Text Descriptors](#)	Shall validate against Text Descriptors schema.

7.1.13 Natural Language Understanding

7.1.13.1 Functions

The Natural Language Understanding (MMC-NLU) AIM receives an input that that might have been generated by a keyboard or by an MMM-ASR AIM and produces a refined text (if the input text was produced by an NNC-ASR AIM, and the Meaning of the input text. The MMC-NLU AIM may also receive the descriptors of an audio-visual scene and the ID of an object:

Receives	Text Object directly input by the Entity.
	Recognised Text from an Automatic Speech Recognition AIM.
	The ID of an Instance.
	The Audio-Visual Scene Descriptors containing the Instance ID.
Refines	Input Text if coming from an Automatic Speech Recognition AIM
Extracts	Meaning (Text Descriptors) from Recognised Text or Entity's Text Object.
Produces	Refined Text.
	Text Descriptors (Meaning).

7.1.13.2 Reference Model

Figure 1 specifies the Reference Model of the Natural Language Understanding (MMC-NLU) AIM.

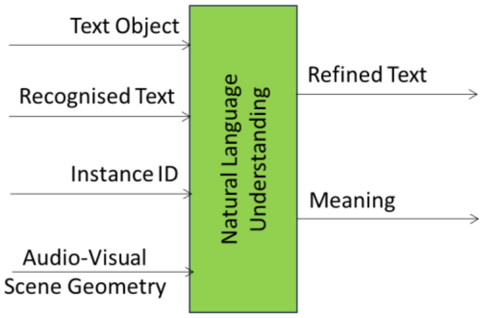

Figure 1 – The Natural Language Understanding (MMC-NLU) AIM Reference Model

7.1.13.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Natural Language Understanding (MMC-NLU) AIM.

Table 1 – I/O Data of the Natural Language Understanding (MMC-NLU) AIM

Input	**Description**
Text Object	Input Text.
Recognised Text Object	Text from the Automatic Speech Recognition AIM.
Instance Identifier	The Identifier of the specific Audio or Visual Object belonging to a level in the taxonomy.

Audio-Visual Scene Geometry	The digital representation of the spatial arrangement of the Visual Objects of the Scene.
Visual Instance Identifier	The Identifier of the specific Visual Object belonging to a level in the taxonomy.
Output	**Description**
Meaning	Descriptors of the Refined Text.
Refined Text Object	The refined version of the Recognised Text from the NLU AIM.

7.1.13.4 JSON Metadata

https://schemas.mpai.community/MMC/V2.4/AIMs/NaturalLanguageUnderstanding.json

7.1.13.5 Profiles

The Profiles of the Natural Language Understanding (MMC-NLU) AIM are specified.

7.1.13.6 Conformance Testing

Table 2 provides the Conformance Testing Method for MMC-NLU 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-NLU AIM

Input	Text Object	Shall validate against Text Object schema. Text Data shall conform with Text Qualifier.
	Recognised Text	Shall validate against Text Object schema. Text Data shall conform with Text Qualifier.
	Instance ID	Shall validate against Instance ID schema.
	Audio-Visual Scene Geometry	Shall validate against AV Scene Descriptors schema.
Output	Refined Text	Shall validate against Text Object schema. Text Data shall conform with Text Qualifier.
	Meaning	Shall validate against Meaning schema.

Table 3 provides an example of MMC-NLU AIM conformance testing.

Table 3 – An example MMC-NLU AIM conformance testing

Input Data	**Data Type**	**Input Conformance Testing Data**
Input Selector	Binary data	All Input Selectors shall conform with Selector.
Text Object	Unicode	All input Text files to be drawn from Text files.
Recognised Text	Unicode	All input Text files to be drawn from Text files.
Output Data	**Data Type**	**Output Conformance Testing Criteria**
Meaning	JSON	All JSON files shall validate against Meaning Schema
Refined Text	Unicode	All Text files produced shall conform with Text.

The four taggings: POS_tagging, NE_tagging, dependency_tagging, and SRL_tagging must be present in the output JSON file of Meaning. Any of the four tagging values may be null.

7.1.14 Personal Status Extraction

7.1.14.1 Functions

The Personal Status Extraction (MMC-PSE) AIM receives the four components of the Personal Status – Text, Speech, Face, and Gesture – or their descriptors and produces the Personal Status. The input selector informs the MMC-PSE whether it should use as input Text, Speech, Face, and Gesture or their descriptors:

Receives	*Text Object* or *Text Descriptors*	
	Text Selector	indicating whether Text or Text Descriptors should be used.
	Speech Object or *Speech Descriptors*	
	Speech Selector	indicating whether Speech or Speech Descriptors should be used.
	Face or *Face Descriptors*	
	Face Selector	indicating whether Face or Face Descriptors should be used.
	Body or *Gesture Descriptors*	
	Body Selector	indicating whether Body or Gesture Descriptors should be used.
Computes and then Interprets	depending on reception of	the Descriptors of a Modality (Text, Speech, or Face).
	Text Descriptors	alternatively, Interprets the received Descriptors and produces Personal Status of the Text Object (PS-Text).

	Speech Descriptors;	alternatively, Interprets the received Descriptors and produces Personal Status of the Speech Object (PS-Speech).
	Face Descriptors	alternatively, Interprets the received Descriptors and produces Personal Status of the Face (PS-Face).
	Gesture Descriptors	alternatively, Interprets the received Gesture Descriptors of the Body.
Multiplexes	The results of the interpretations.	
Produces	Entity's Personal Status	

7.1.14.2 Reference Model

Figure 1 depicts the Reference Model of the Personal Status Extraction (MMC-PSE) AIM.

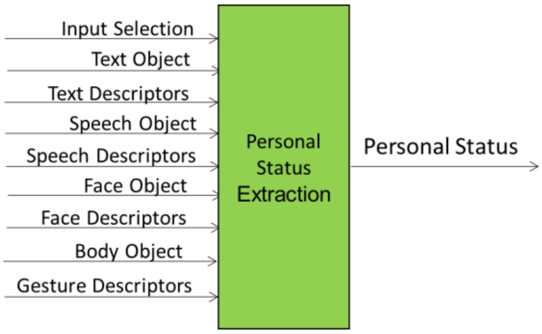

Figure 1 – The Personal Status Extraction Composite (MMC-PSE) AIM Reference Model

7.1.14.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Personal Status Extraction (MMC-PSE) AIM.

Table 1 – I/O Data of the Personal Status Extraction (MMC-PSE) AIM

Input data	**From**	**Description**
Input Selector	An external signal	Media or Descriptors Selector
Text Object	Keyboard or AIM	Text or Recognised Text.
Text Descriptors	An upstream AIM	Functionally equivalent to Text Description.
Speech Object	Microphone/upstream AIM	Speech of Entity.
Speech Descriptors	An upstream AIM	Functionally equivalent to Speech Description.
Face Visual Object	Visual Scene Description	The face of the Entity.
Face Descriptors	An upstream AIM	Functionally equivalent to Face Description.

Body Visual Object	Visual Scene Description	The body of the Entity.
Gesture Descriptors	An upstream AIM	Functionally equivalent to Body Description.
Output data	**To**	**Description**
Personal Status	A downstream AIM	For further processing

7.1.14.4 SubAIMs

A Personal Status Extraction AIM instance can be implemented as a Composite AIM with different degrees of composition. The most extended composition if depicted by Figure 2

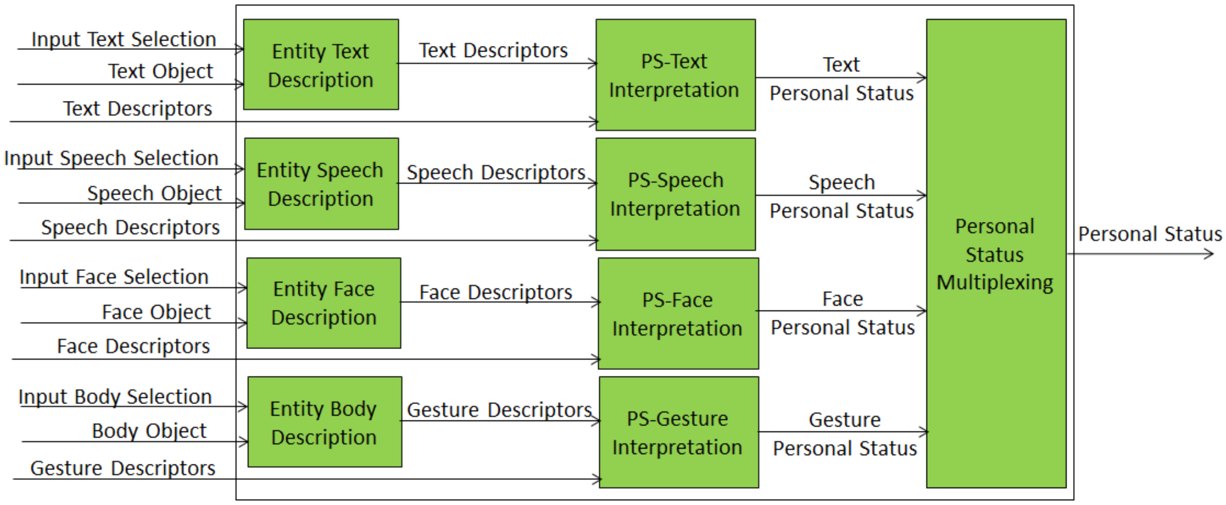

Figure 2 - The version of Personal Status Extraction AIM with the highest level of composition.

Table 2 gives the AIMs and their JSON Metadata of MMC-PSE.

Table 2 - AIMs and JSON Metadata

AIMs	AIMs	AIM Names	JSON
MMC-PSE		Personal Status Extraction	X
	MMC-ETD	Entity Text Description	X
	MMC-ESD	Entity Speech Description	X
	PAF-EFD	Entity Face Description	X
	PAF-EBD	Entity Body Description	X
	MMC-PTI	PS-Text Interpretation	X
	MMC-PSI	PS-Speech Interpretation	X
	PAF-PFI	PS-Face Interpretation	X
	PAF-PGI	PS-Gesture Interpretation	X
	MMC-PMX	Personal Status Multiplexing	X

7.1.14.5 JSON Metadata

https://schemas.mpai.community/MMC/V2.4/AIMs/PersonalStatusExtraction.json

7.1.14.6 Profiles

The Profiles of Personal Status Extraction are specified.

7.1.14.7 Conformance Testing

Table 3 provides the Conformance Testing Method for MMC-PSE AIM as a Basic AIM. Conformance Testing of the individual AIMs of the MMC-PSE 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 that refers to a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 3 – Conformance Testing Method for MMC-PSE AIM

Input	Text Object or	Shall validate against Text Object schema. Text Data shall conform with Text Qualifier.
	Text Descriptors	Shall validate against Text Descriptors schema.
	Text Selector	Shall validate against Text Selector schema.
	Speech Object or	Shall validate against Speech Object schema. Speech Data shall conform with Speech Qualifier.
	Speech Descriptors	Shall validate against Speech Descriptors schema.
	Speech Selector	Shall validate against Speech Selector schema.
	Face Visual Object or	Shall validate against Visual Object schema. Visual Data shall conform with Visual Qualifier.
	Face Descriptors	Shall validate against Face Descriptors schema.
	Face Selector	Shall validate against Face Selector schema.
	Body Visual Object	Shall validate against Visual Object schema. Visual Data shall conform with Visual Qualifier.
	Gesture Descriptors	Shall validate against Gesture Descriptors schema.
	Body Selector	Shall validate against Body Selector schema.
Output	Entity Personal Status	Shall validate against Personal Status schema.

7.1.15 Personal Status Multiplexing

7.1.15.1 Functions

The Personal Status Multiplexing (MMC-PSM) AIM multiplexes the components elements of a Personal Status instance - Text Personal Status, Speech Personal Status, Face Personal Status, and Gesture Personal Status - into a Personal Status:

Receives	*PS-Text*	Personal Status of Text

	PS-Speech	Personal Status of Speech
	PS-Face	Personal Status of Face
	PS-Gesture	Personal Status of Gesture
Produces	*Personal Status*	Multiplexed Personal Status

7.1.15.2 Reference Model

Figure 1 depicts the Reference Model of the Personal Status Multiplexing (MMC-PSM) AIM.

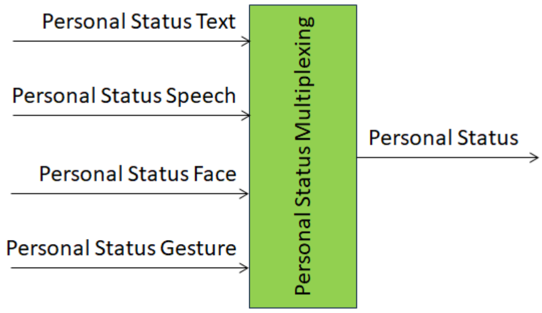

Figure 1 – The Personal Status Multiplexing (MMC-PSM) AIM Reference Model

7.1.15.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Personal Status Multiplexing (MMC-PSM) AIM.

Table 1 – I/O Data of the Personal Status Multiplexing (MMC-PSM)

Input	**Description**
Text Personal Status	Personal Status of Text Object.
Speech Personal Status	Personal Status of Speech Object.
Face Personal Status	Personal Status of Face Object.
Gesture Personal Status	Personal Status of Gesture conveyed by Body Object.
Output	**Description**
Personal Status	Personal Status of Machine.

7.1.15.4 JSON Metadata

https://schemas.mpai.community/MMC/V2.4/AIMs/PersonalStatusMultiplexing.json

7.1.15.5 Conformance Testing

Table 2 provides the Conformance Testing Method for MMC-PSM 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-PSM AIM

Input	Text Personal Status	Shall validate against Text Personal Status schema.
	Speech Personal Status	Shall validate against Speech Personal Status schema.
	Face Personal Status	Shall validate against Face Personal Status schema.
	Gesture Personal Status	Shall validate against Gesture Personal Status schema.
Output	Personal Status	Shall validate against Personal Status schema.

7.1.16 PS-Speech Interpretation

7.1.16.1 Functions

The PS-Speech Interpretation (MMC-PSI) AIM uses input speech descriptors to extract the Personal Status from it:

Receives	*Speech Descriptors*	to be interpreted.
Produces	*PS-Speech*	The Personal Status of the Speech Modality.

7.1.16.2 Reference Model

Figure 1 depicts the Reference Model of the PS-Speech Interpretation (MMC-PSI) AIM.

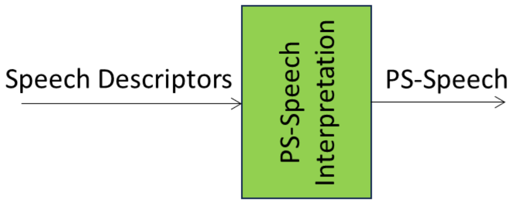

Figure 1 The PS-Speech Interpretation (MMC-PSI) AIM Reference Model

7.1.16.3 Input/Output Data

Table 1 specifies the Input and Output Data of the PS-Speech Interpretation (MMC-PSI) AIM.

Table 1 – I/O Data of the PS-Speech Interpretation (MMC-PSI) AIM

Input	Description
Speech Descriptors	Descriptors of Speech
Output	**Description**
Speech Personal Status	Personal Status of Speech

7.1.16.4 JSON Metadata

https://schemas.mpai.community/MMC/V2.4/AIMs/PSSpeechInterpretation.json

7.1.16.5 Profiles

No Profiles.

7.1.16.6 Conformance Testing

Table 2 provides the Conformance Testing Method for MMC-PSI 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-PSI AIM

Input	Speech Descriptors	Shall validate against Speech Descriptors schema
Output	Speech Personal Status	Shall validate against Speech Personal Status schema

7.1.17 PS-Text Interpretation

7.1.17.1 Functions

The PS-Text Interpretation (MMC-PTI) AIM uses input text descriptors to extract the Personal Status from it:

Receives	*Text Descriptors*	Either from Text Description or as a direct input to PS-Text Interpretation.
Produces	*PS-Text*	the Personal Status of the Text Modality.

7.1.17.2 Reference Model

Figure 1 depicts the Reference Model of the PS-Text Interpretation (MMC-PRI) AIM.

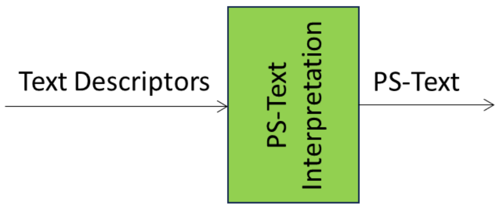

Figure 1 – The PS-Text Interpretation (MMC-PRI) AIM Reference Model

7.1.17.3 Input/Output Data

Table 1 specifies the Input and Output Data of the PS-Text Interpretation (MMC-PRI) AIM.

Table 1 – I/O Data of the PS-Text Interpretation (MMC-PRI) AIM

Input	Description
Text Descriptors	Descriptors of Text Data
Output	**Description**
Text Personal Status	Personal Status of Text Data

7.1.17.4 JSON Metadata

https://schemas.mpai.community/MMC/V2.4/AIMs/PSTextInterpretation.json

7.1.17.5 Conformance Testing

Table 2 provides the Conformance Testing Method for MMC-PRI 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-PRI AIM

Input	Text Descriptors	Shall validate against Text Descriptors schema
Output	Text Personal Status	Shall validate against Text Personal Status schema

7.1.18 Speaker Identity Recognition

7.1.18.1 Functions

The Speaker Identity Recognition (MMC-SIR) AIM receives an input speech and produces the identifier of the Entity producing the input speech. the (MMC-SIR) AIM may also receive auxiliary text connected with the input speech, the start and end time during which the identifier of the speaker Entity is requested, the Speech Overlap data type signaling if more than one speaker has produces the input speech and the Geometry of the Speech Scene:

Receives	*Auxiliary Text*	Text related to the Speech.
	Speech Object	Speech of which the Speaker is requested.
	Speech Time	Time during whose duration Speaker ID is requested.
	Speech Overlap	Data signaling which parts of Speech Data have overlapping speech.
	Speech Scene Geometry	Disposition of Speech Data of the scene where the Speech whose speaker is to be identified is located.
Produces	*Speaker Identifier*	ID of speaker.

7.1.18.2 Reference Model

The Reference Architecture of Speaker Identity Recognition (MMC-SIR) is depicted in Figure 1.

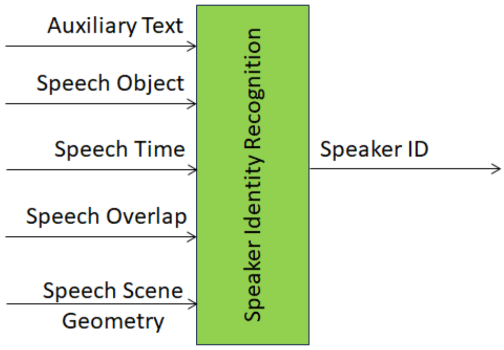

Figure 1 – The Speaker Identity Recognition (MMC-SIR) AIM

7.1.18.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Speaker Identity Recognition (MMC-SIR) AIM.

Table 1 – I/O Data of the Speaker Identity Recognition (MMC-SIR) AIM

Input	**Description**
Auxiliary Text Object	Text with content related to Speaker ID.
Speech Object	Speech Object emitted by the Speaker.
Speech Time	The start and end time of the Speech.
Speech Overlap	Information about overlapping Speech.
Speech Scene Geometry	Information about Speech Object location.
Output	**Description**
Speaker Identifier	The Visual Descriptors of the Visual Scene.

7.1.18.4 JSON Metadata

https://schemas.mpai.community/MMC/V2.4/AIMs/SpeakerIdentityRecognition.json

7.1.18.5 Reference Software

7.1.18.5.1 Disclaimers

1. This MMC-SIR Reference Software Implementation is released with the BSD-3-Clause licence.
2. The purpose of this MMC-SIR Reference Software is to show a working Implementation of MMC-SIR, 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.

7.1.18.5.2 Guide to the MMC-SIR code

MMC-SIR performs speaker verification with a pretrained ECAPA-TDNN model; that is, it identifies the speaker of each speech segment by comparison with a dataset consisting of short clips of human speech.

The MMC-SIR 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: commands for cloning https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb

Library: https://github.com/speechbrain/speechbrain

7.1.18.5.3 Acknowledgements

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

7.1.18.6 Conformance Testing

Table 2 provides the Conformance Testing Method for MMC-SIR 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-SIR AIM

Input	Text Object	Shall validate against Text Object schema. Auxiliary Text Data shall conform with Text Qualifier.
	Speech Object	Shall validate against Speech Object schema. Speech Data shall conform with Speech Qualifier.
	Speech Time	Shall validate against Time schema.
	Speech Overlap	Shall validate against Speech Overlap schema. Speech Data shall conform with Speech Qualifier.
	Speech Scene Geometry	Shall validate against Speech Scene Geometry schema.
Output	Speaker Identifier	Shall validate against Instance ID schema.

7.1.18.7 Performance Assessment

Performance Assessment of an MMC-SIR AIM Implementation shall be performed using a dataset of speech segments all in the same language, for each segment of which the Identity of the Speaker is provided with reference to a Taxonomy.
The Performance Assessment Report of an MMC-SIR AIM Implementation shall include:

1. The Identifier of the MMC-SIR AIM.
2. The Identifier of the speech segment dataset.
3. The language of the speech segment dataset.
4. The Taxonomy of Speaker Identifiers.
5. The Performance of the MMC-SIR AIM expressed as the Accuracy of the Identifiers provided by the MMC-SIR AIM computed on all speech segments of the dataset referenced in 2.

7.1.19 Text and Speech Translation

7.1.19.1 Functions

The Text and Speech Translation (MMC-TST) AIM receives an input text or an input speech and languages preferences informing about the language of the input text or speech and the target language of the output text or speech and produces, independently of whether the input is text or speech a text or speech in the language indicated in the language preferences. The different selection are signaled by the input selector:

Receives	Selector	To choose between:
		- The AIM output should be Text or Speech.
		- The output Speech should retain the input Speech Features.
	Language Preferences	as requested input and output language.
	Personal Status.	Use of Personal Status
	Text.	Use of Text
	Speech.	Use of Speech
Performs	A subset of) the following:	
	Conversion of input Speech	Into Text.
	Translation of Text	To the target language.
	Extraction of Features	From Speech.
	Conversion of Text	Into Speech adding the Input Speech's Features.
Produces	Translated Text.	Depends of Selector.
.	Translated Speech	Depends of Selector.

7.1.19.2 Reference Model

Figure 1 depicts the Reference Model of the Text-and-Speech Translation Composite (MMC-TST) AIM.

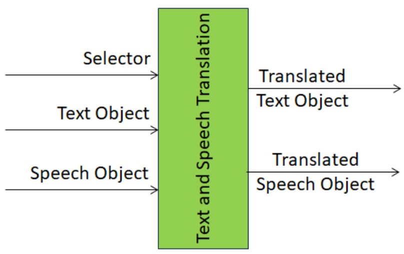

Figure 1 – Text-and-Speech Translation (MMC-TST) AIM Reference Model

7.1.19.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Text-to-Text Translation (MMC-TST) AIM.

Table 1 – I/O Data of the Text-and-Speech Translation (MMC-TST) AIM

Input	Semantics
Selector	Signals: 1. Whether the input is Text or Speech 2. Whether the input Speech features are preserved in the output Speech. 3. The Input and output languages.
Speech Object	Speech produced in input language by a human desiring translation into output language
TextObject	Alternative textual source information to be translated into and pronounced in output language depending on the value of Input Selection.
Output	**Description**
Translated SpeechObject	Speech in input language translated into output language preserving the Input Speech features in the Output Speech, depending on Selector.
Translated TextObject	Text of Input Speech or Input Text translated into output language, depending on Selector.

7.1.19.4 SubAIMs

Text and Speech Translation is a Composite AIM whose Reference Model is depicted in Figure 2.

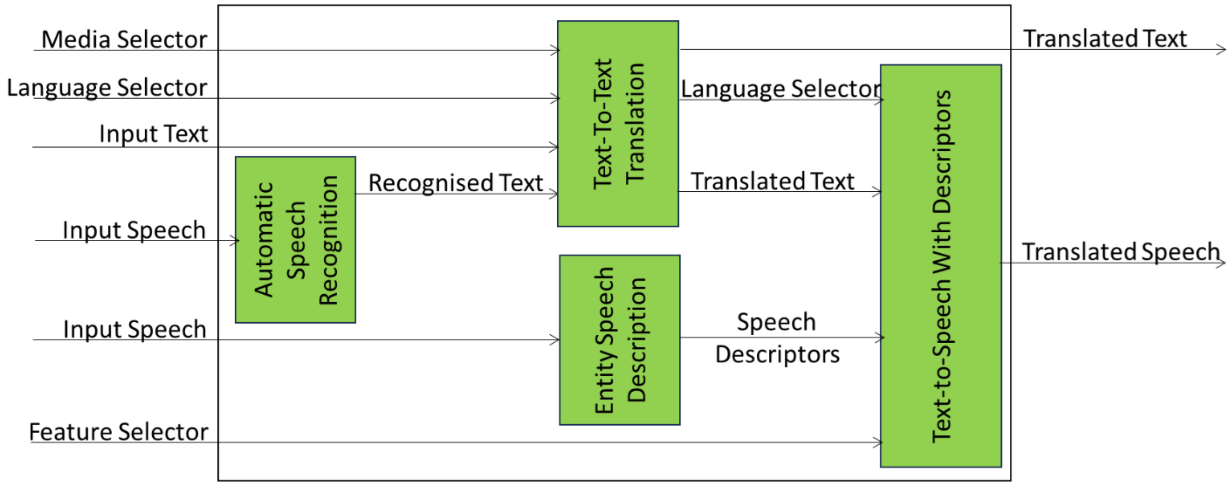

Figure 2 – Text-and-Speech Translation Composite (MMC-TST) AIM

Table 2 - AIMs of Text-and-Speech Translation Composite (MMC-TST) AIM

AIW	AIMs	AIM Names	JSON
MMC-TST		Text-and-Speech Translation	X
	MMC-ASR	Automatic Speech Recognition	X
	MMC-TTT	Text-to-Text Translation	X
	MMC-ISD	Entity Speech Description	X
	MMC-DTS	Descriptors Text-to-Speech	X

7.1.19.5 JSON Metadata

https://schemas.mpai.community/MMC/V2.4/AIMs/TextAndSpeechTranslation.json

7.1.19.6 Profiles

The Profiles of Text and Speech Translation are specified.

7.1.19.7 Conformance Testing

Table 3 provides the Conformance Testing Method for MMC-TST AIM as a Basic AIM. Conformance Testing of the individual AIMs of the MMC-TST 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 3 – Conformance Testing Method for MMC-TST AIM

Input	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.
Output	Translated Text Object	Shall validate against Text Object. Text Data shall conform with Text Qualifier.
	Translated Speech Object	Shall validate against Speech Object. Speech Data shall conform with Speech Qualifier.

Important note. This Conformance Testing Specification does not provide methods and datasets to Test the Conformance of the individual Speech Feature Extraction and Text-To-Speech Basic AIMs, only of their Descriptors Speech Translation Composite AIMs.

Table 4 provides an example of MMC-TST AIM conformance testing.

Table 4 – An example MMC-TST AIM conformance testing

Input Data	**Data Type**	**Input Conformance Testing Data**
Input Selector	Selector	All Input Selectors to conform with Selector.
Requested Language	Selector	All Language Selectors to be drawn from Language Codes.
Input Text	Unicode	All input Text files shall be drawn from Text files.
Input Speech	.wav	All input Text files shall be drawn from Speech files.
Output Data	**Data Type**	**Conformance Test**
Machine Text	Unicode	All Text files produced shall conform with Text files.
Machine Speech	.wav	All Speech files produced shall conform with Speech files.

7.1.20 Text-To-Speech

7.1.20.1 Functions

The Text-To-Speech (MMC-TTS) AIM receives an input text and produces a synthetic speech version of it. The MMC-TTT AIM may also receive the personal Status to be used in the synthetic speech and a Speech Model:

Receives	Text Object	Input Text
	Personal Status	to be contained in the Synthesised Speech Object.
	Speech Model	used by AIM depending on Profile.
Feeds	Text Object and Personal Status	to Speech Model.
Produces	Synthesised Speech Object	output of AIM.

7.1.20.2 Reference Model

Figure 1 specifies the Reference Model of the Text-To-Speech (MMC-TTS) AIM.

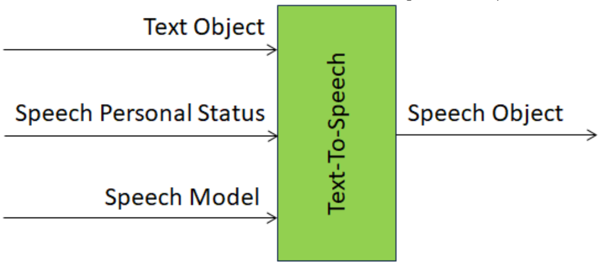

Figure 1 – The Text-To-Speech AIM Reference Model

7.1.20.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Automatic Speech Recognition AIM.

Table 1 – I/O Data of the Automatic Speech Recognition AIM

Input	Description
Text Object	Input Text.
Personal Status	Input Personal Status of the Speech Modality.
Speech Model	NN Model used to produce Speech from Text and Personal Status.
Output	**Description**
Speech Object	Output of the Text-To-Speech AIM,

7.1.20.4 JSON Metadata

https://schemas.mpai.community/MMC/V2.4/AIMs/TextToSpeech.json

7.1.20.5 Profiles

The Text-To-Speech Profiles are specified.

7.1.20.6 Reference Software

7.1.20.6.1 Disclaimers

1. The purpose of this MMC-TTS Reference Software is to provide a working Implementation of MMC-TTS, not to provide a ready-to-use product.
2. MPAI disclaims the suitability of the Software for any other purposes and does not guarantee that it is secure.
3. 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.

7.1.20.6.2 Guide to the MMC-TTS code

Use of this AI Module is for developers who are familiar with Python and downloading models from HuggingFace.
A wrapper for the speech5 NN Module

1. Manages input files and parameters: Text Object
2. Executes the BLIP Module to perform the Speech Recognition on each individual pair of Text and Visual Object.
3. Outputs Speech Object as answer.

The MMC-TTS Reference Software is found at the MPAI-NNW gitlab site. It contains:

1. The python code implementing the AIM
2. Required libraries are: pytorch, transformers (HuggingFace), datasets (HuggingFace), and soundfile.

7.1.20.6.3 Acknowledgements

This version of the MMC-TTS Reference Software has been developed by the MPAI *Neural Network Watermarking* Development Committee (NNW-DC).

7.1.20.7 Conformance Testing

Table 2 provides the Conformance Testing Method for MMC-TTS 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-TTS AIM

Input	Text Object	Shall validate against Text Object schema. Text Data shall conform with Text Qualifier.
	Personal Status	Shall validate against Personal Status schema.
	Speech Model	Shall validate against Machine Learning Model schema. Machine Learning Model Data shall conform with Machine Learning Model Qualifier.
Output	Synthesised Speech Object	Shall validate against Speech Object schema. Speech Data shall conform with Speech Qualifier.

Table 3 provides an example of MMC-TTS AIM conformance testing.

Table 3 – An example MMC-TTS AIM conformance testing

E	**Data Type**	**Input Conformance Testing Data**
Machine Text	Unicode	All input Text files to be drawn from Text files.
Machine Emotion	JSON	All input JSON Emotion files to be drawn from Emotion JSON Files
Output Data	**Data Type**	**Output Conformance Testing Criteria**
Machine Speech	.wav	All Speech files produced shall conform with Speech.

7.1.21 Text-to-Text Translation

7.1.21.1 Functions

The Text-to-Text Translation (MMM-TTT) AIM receives an input text and produces a text in a different language. The MMM-TTT AIM may also receive the Meaning of the input text:

Receives	*Selector*	Determining the input and target language.
	Text Object	Text to be translated.
	Meaning	Input Text Meaning.
Produces	*Translated Text*	Output Translates Text.

7.1.21.2 Reference Model

Figure 1 depicts the Reference Model of the Text-to-Text Translation AIM.

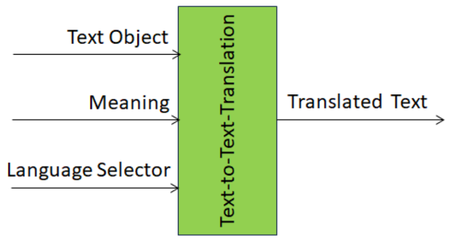

Figure 1 – Text-to-Text Translation AIM Reference Model

7.1.21.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Text-to-Text Translation AIM.

Table 1 – I/O Data of the Text-to-Text Translation AIM

Input	**Description**
Text Object	Input Text Object.
Meaning	Meaning of Input Text
Language Selector	Input and target Language.
Output	**Description**
Translated Text Object	Translation of Text (or Refined Text).

7.1.21.4 JSON Metadata

https://schemas.mpai.community/MMC/V2.4/AIMs/TextToTextTranslation.json

7.1.21.5 Conformance Testing

Table 2 provides the Conformance Testing Method for MMC-TTT 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-TTT AIM

Input	Language Selector	Shall validate against Language Selector schema.
	Text Object	Shall validate against Text Object schema. Text Data shall conform with Text Qualifier.
	Meaning	Shall validate against Meaning schema.
Output	Translated Text Object	Shall validate against Text Object schema. Text Data shall conform with Text Qualifier.

Table 3 provides an example of MMC-TTT AIM conformance testing.

Table 3 – An example MMC-TTT AIM conformance testing

Input Data	**Data Type**	**Input Conformance Testing Data**
Input Text	Unicode	All input Text files to be drawn from Text files.
Output Data	**Data Type**	**Output Conformance Testing Criteria**
Translated Text	Unicode	All Text files produced shall conform with Text.

7.1.21.6 Performance Assessment

Performance Assessment of an MMC-TTT AIM Implementation shall be performed using a dataset of text sentences in a given language. Each text sentence shall have at least one translated text.
The Performance Assessment Report of an MMC-TTT AIM Implementation shall include:
1. The Identifier of the MMC-TTT AIM.
2. The Identifier of the dataset of text sentences.
3. The name of the input and output languages and their ISO 639 Set 3 three-letter code.

4. The number of text sentences in the data set and the average number of translated texts per input text.
5. The maximum value N of n-grams used.
6. The BLEU Score of the MMC-TTT AIM, defined as the Arithmetic Mean of the individual BLEU Scores computed over the dataset, where each BLEU Score is the product of the Brevity Penalty and the Geometric Mean Precision, and where:
    1. The *Brevity Penalty* of a candidate translation of length *c* to a reference translation of length *r* is min $(1,e^{(1-r/c)})$.
    2. The *Sentence Precision* of a set of N n-grams is $\exp(\sum_{n=1,N}\log(p_n)/N$, where p_i is the precision of the i-th n-gram.

7.1.22 Audio Scene Description

7.1.22.1 Functions

The Audio Scene Description (OSD-ASD) AIM receives Audio Objects, the Descriptors of the Scene the Objects belong to, and their Space-Time information as inputs and produces the Descriptors of a Scene that is composed of Audio Objects and Scenes as output. The OSD-ASD AIM may also receive an Alert conveying information on potential anomalies in the input Audio Objects:

Receives	Space-Time	of the input Objects having the same time base.
	Audio Objects	individual Audio Objects.
	Scene Descriptors	Scene to Objects belong to.
Integrates	Space-Time and 3D Model Object	with Scene Descriptors.
Produces	Audio Scene Descriptors	Output#1 of AIM
	Alert	Output#2 of AIM signalling potential anomalies in Object.

7.1.22.2 Reference Model

The Reference Architecture is depicted in Figure 1.

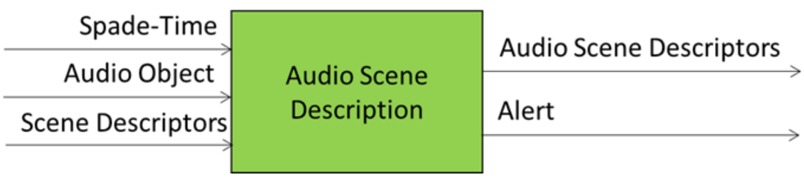

Figure 1 – The Audio Scene Description (OSD-ASD) AIM

7.1.22.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio Scene Description (OSD-ASD) AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Audio Scene Description (OSD-ASD) AIM

Input	Description
Space-Time	Space-Time of input Objects.
Audio Objects	Input Objects.
Scene Descriptors	Input Scene Descriptors.
Output	**Description**
Audio Scene Descriptors	The output Audio Scene Descriptors.
Alert	Data signalling potential anomalies in Object.

7.1.22.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.4/AIMs/SceneDescription.json

7.1.22.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-3SD 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 OSD-3SD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Audio Objects	Shall validate against Audio Object schema. Media-specific Data shall conform with their Qualifiers.
	Scene Descriptors	Shall validate against Scene Descriptors schema.
Produces	Audio Scene Descriptors	Shall validate against Audio Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.1.23 Audio-Visual Alignment

7.1.23.1 Functions

The Audio-Visual Alignment (OSD-AVA) AIM provides the Descriptors of an Audio-Visual Scene whose Audio Objects, Speech Objects, 3D Model Objects, and Visual Objects have compatible Identifiers if they have the same Position.

Receives	*Speech Scene Descriptors*	Descriptors of potentially present Speech Scene.
	Audio Scene Descriptors	Descriptors of potentially present Audio Scene.
	Visual Scene Descriptors	Descriptors of potentially present Visual Scene.
	3D Model Scene Descriptors	Descriptors of potentially present 3D Model Scene.

Aligns	Speech, Audio, and Visual Objects	Sharing the same Spatial Attitude
Produces	*Audio-Visual Scene Descriptors*	Where Speech Objects, Audio Objects, 3D Model Objects, and Visual Objects have compatible Identifiers if they have the same Spatial Attitude.

7.1.23.2 Reference Model

Figure 1 specifies the Reference Model of the Audio-Visual Alignment (OSD-AVA) AIM.

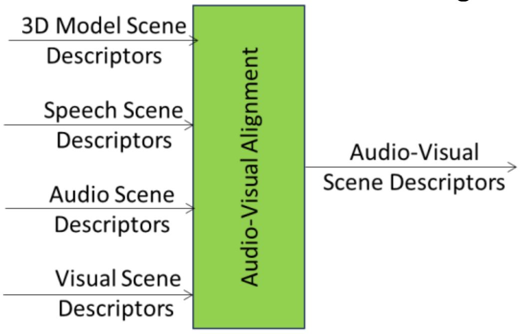

Figure 1 - Reference Model of the Audio-Visual Alignment (OSD-AVA) AIM

7.1.23.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio-Visual Alignment (OSD-AVA) AIM.

Table 1 – I/O Data of the Audio-Visual Alignment AIM

Input	**Description**
Speech Scene Descriptors	The IDs and the geometry of the Speech Objects of the Scene.
Audio Scene Descriptors	The IDs and the geometry of the Audio Objects of the Scene.
Visual Scene Descriptors	The IDs and the geometry of the Audio Objects of the Scene.
3D Model Scene Descriptors	The Descriptors of the 3D Model Scene.
Output	**Description**
Audio-Visual Scene Descriptors	The IDs and the geometry of the Audio, Speech, 3D Model, Visual and Audio-Visual Objects of the Scene.

7.1.23.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.4/AIMs/AudioVisualAlignment.json

7.1.23.4.1 Disclaimers

1. This OSD-AVA Reference Software Implementation is released with the BSD-3-Clause licence.
2. The purpose of this Reference Software is to show a working Implementation of OSD-AVA, not to provide a ready-to-use product.
3. MPAI disclaims the suitability of this Reference 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.

7.1.23.4.2 Guide to OSD-AVA code

OSD-AVA arranges the output Visual Objects and Speech Objects with the corresponding Time information: scene cuts/transitions and speakers' turns. Each Object is bounded by two adjacent times from a list of unique times that are either 1) scene cuts/transitions or 2) starts and ends of speakers' turns.
Use of this Reference Software for the OSD-AVA AI Module is for developers who are familiar with Python, Docker, and RabbitMQ.
OSD-AVA computes segments as unique intervals from scene bounds and from speech segments. Moreover, OSD-AVA outputs visual objects and speech objects.
The OSD-AVA 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.

7.1.23.4.3 Acknowledgements

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

7.1.23.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-AVA 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 OSD-AVA AIM

Receives	Speech Scene Descriptors	Shall validate against Speech Scene Descriptors schema
	Audio Scene Descriptors	Shall validate against Audio Scene Descriptors schema
	Visual Scene Descriptors	Shall validate against Visual Scene Descriptors schema
	3D Model Scene Descriptors	Shall validate against 3D Model Scene Descriptors schema
Produces	Audio-Visual Scene Descriptors	Shall validate against AV Scene Descriptors schema

7.1.23.6 Performance Assessment

Performance Assessment of an OSD-AVA AIM Implementation shall be performed using a dataset of scenes containing Audio and/or Speech and Visual objects.
The Performance Assessment Report of an OSD-AVA AIM Implementation shall include:

1. The Identifier of the OSD-AVA AIM whose Performance is being Assessed.
2. The Identifier of the scene dataset used which include the identifiers of the aligned objects.
3. The data type of the scenes: analogue, digital, without or with separated objects.
4. The Performance of the OSD-AVA AIM expressed as the number of times the OSD-AVA AIM being Assessed for Performance:
    - Correctly identifies as aligned the objects that the data set declares as aligned divided by the total number of aligned objects (Truly aligned objects).
    - Incorrectly identifies as aligned the object that the dataset declares aligned in the dataset divided by the total number of aligned objects (Falsely aligned objects).

- Incorrectly identifies as non-aligned object that are declared aligned in the dataset referenced in 2 divided by the total number of aligned objects (Missed aligned objects).

7.1.24 Audio-Visual Event Description

7.1.24.1 Functions

The Audio-Visual Event Description (OSD-MED) AIM produces the Descriptors of an Audio-Visual Event from a sequence of Audio-Visual Scene Descriptors:

Receives	*Audio-Visual Scene Descriptors*.	A sequence.
Produces	*Audio-Visual Event Descriptors*	

7.1.24.2 Reference Model

The Audio-Visual Event Description (OSD-MED) AIM Reference Model is depicted in Figure 1.

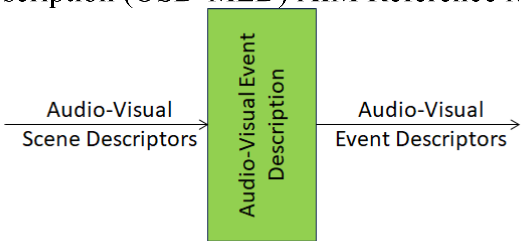

Figure 1 – The Audio-Visual Event Description (OSD-MED) AIM Reference Model

7.1.24.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio-Visual Event Description AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Audio-Visual Event Description (OSD-MED) AIM

Input	Description
Audio-Visual Scene Descriptors	Sequence of Audio-Visual Scene Descriptors.
Output	**Description**
Audio-Visual Event Descriptors	The Audio-Visual Event Descriptors of the Audio-Visual Scene.

7.1.24.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.4/AIMs/AudioVisualEventDescription.json

7.1.24.5 Reference Software

1. This OSD-MED Reference Software Implementation is released with the BSD-3-Clause licence.
2. The purpose of this Reference Software is to show a working Implementation of OSD-MED, 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.

7.1.24.5.1 Guide to the OSD-AVE code

OSD-MED arranges the audio-visual scene descriptors from OSD-AVS into Audio-Visual Event Descriptors.
Use of this Reference Software for the OSD-MED AI Module is for developers who are familiar with Python, Docker, and RabbitMQ.
The OSD-MED 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.

7.1.24.5.2 Acknowledgements

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

7.1.24.6 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-MED 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 OSD-MED AIM

Receives	Audio-Visual Scene Descriptors	Shall validate against AV Scene Descriptors schema
Produces	Audio-Visual Event Descriptors	Shall validate against AV Event Descriptors schema

7.1.25 Audio-Visual Scene Demultiplexing

7.1.25.1 Functions

The Audio-Visual Scene Demultiplexing (OSD-SDX) receives Audio-Visual Scene Descriptors and extracts the component elements - Audio Scene Geometry, Speech Scene Geometry, Visual Scene Geometry, Audio Objects, Speech Objects, and Visual Objects - from Audio-Visual Scene Descriptors:

Receives	*Audio-Visual Scene Descriptors*
Demultiplexes	*Audio-Visual Scene Descriptors*
Produces	*Speech Scene Geometry*
	Audio Scene Geometry
	Visual Scene Geometry
	Speech Objects
	Audio Objects
	Visual Objects

7.1.25.2 Reference Model

Figure 1 depicts the Reference Model of the Audio-Visual Scene Demultiplexing (OSD-SDX) AIM.

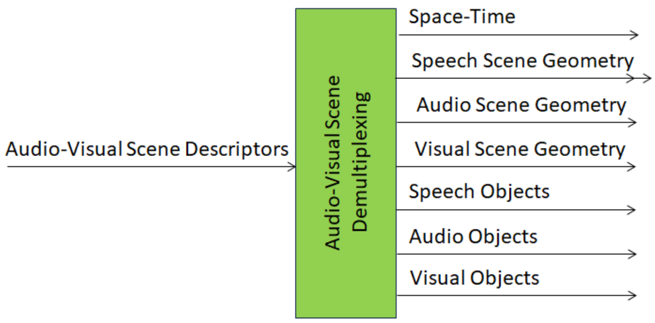

Figure 1 – Audio-Visual Scene Demultiplexing (OSD-SDX) AIM Reference Model

7.1.25.3 Input/Output Data

Table 1 specifies the Input and Output Data of the of the Audio-Visual Scene Demultiplexing (OSD-SDX) AIM.

Table 1 – I/O Data of the Audio-Visual Scene Demultiplexing (OSD-SDX) AIM

Input	Description
Audio-Visual Scene Descriptors	The Descriptors of the Audio-Visual Scene.
Output	**Description**
Space-Time	Space-Time information of the Audio-Visual Scene
Speech Scene Geometry	The Descriptors of the Speech Scene.
Audio Scene Geometry	The Descriptors of the Audio Scene.
Visual Scene Geometry	The Descriptors of the Visual Scene.
Audio Object	The Audio Objects in the Scene.
Speech Object	The Speech Objects in the Scene.
Visual Object	The Visual Objects in the Scene.

7.1.25.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.4/AIMs/AudioVisualSceneDemultiplexing.json

7.1.25.5 Conformance Testing

Table 2 provides the Conformance Testing Method for the OSD-SDX AIM as a Basic 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 OSD-SDX AIM

Receives	Audio-Visual Scene Descriptors	Shall validate against AV Scene Descriptors schema
Produces	Speech Scene Geometry	Shall validate against Speech Scene Geometry schema
	Audio Scene Geometry	Shall validate against Audio Scene Geometry schema
	Visual Scene Geometry	Shall validate against Visual Scene Geometry schema
	Speech Objects	Shall validate against Speech Objects schema Speech Data shall conform with Qualifier
	Audio Objects	Shall validate against Audio Objects schema Audio Data shall conform with Qualifier
	Visual Objects	Shall validate against Visual Objects schema Visual Data shall conform with Qualifier

7.1.26 Audio-Visual Scene Description

7.1.26.1 Functions

The Audio-Visual Scene Description (OSD-MSD) AIM receives Audio-Visual Objects, the Descriptors of the Scene the Objects belong to, and their Space-Time information as inputs and produces the Descriptors of a Scene that is composed of Audio-Visual Objects and Scenes as output. The OSD-MSD AIM may also receive an Alert conveying information on potential anomalies in the input Audio-Visual Objects:

Receives	*Space-Time*	Of output Audio-Visual Scene Descriptors,
	Speech Objects	
	Audio Objects	
	Visual Objects	
	Audio-Visual Scene Descriptors	Of Scene to be augmented.
Augments	*Audio-Visual Scene Descriptors*	
Produces	*Audio-Visual Scene Descriptors*	

7.1.26.2 Reference Model

Figure 1 specified the Reference Model of Audio-Visual Scene Description (OSD-AVS) aim.

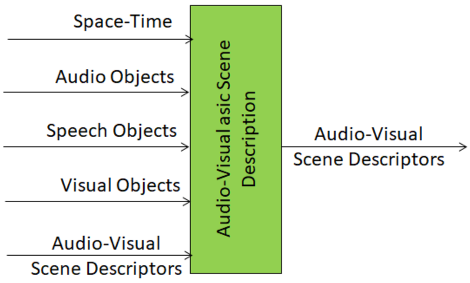

Figure 1 – The Audio-Visual Scene Description (OSD-AVS) AIM

7.1.26.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio-Visual Scene Description (OSD-AVS) AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Audio-Visual Scene Description (OSD-AVS) AIM

Input	Description
Space-Time	Space-Time information of output Audio-Visual Scene Descriptors
Speech Object	Speech Object
Audio Objects	Audio Objects.
Visual Objects	Visual Objects.
Audio-Visual Scene Descriptors	The Audio-Visual Descriptors of the Scene part of the target Audio-Visual Scene.
Output	**Description**
Audio-Visual Scene Descriptors	The Audio-Visual Descriptors of the Scene.

7.1.26.4 SubAIMs

Figure 2 specified the Reference Model of Audio-Visual Scene Description (CAE-ASD) Composite AIM.

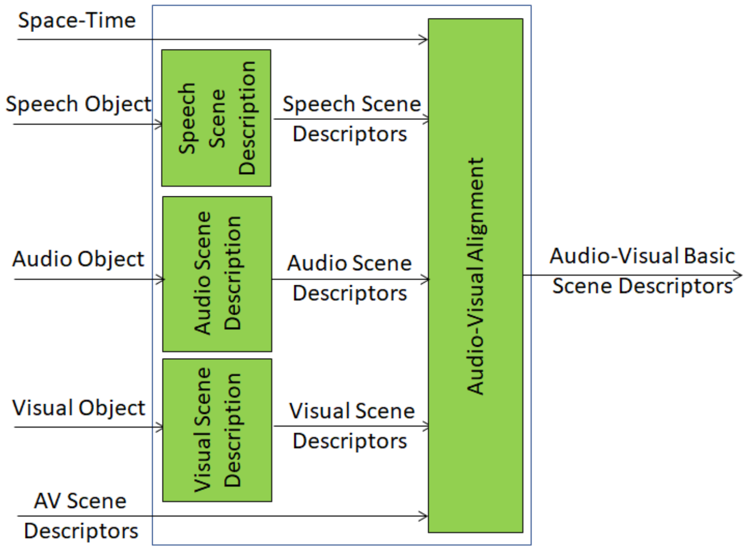

Figure 2 – The Audio-Visual Scene Description (OSD-AVS) Composite AIM

Table 2 provides the links to the specifications of the OSD-AVS AIMs.

Table 2 – AIMs of the Audio-Visual Scene Description (OSD-AVS) Composite AIM

AIMs	Names	JSON
OSD-SSD	Speech Scene Description	X
OSD-ASD	Audio Scene Description	X
OSD-VSD	Visual Scene Description	X
OSD-AVA	Audio-Visual Alignment	X

7.1.26.5 JSON Metadata

http://schemas.mpai.community/OSD/V1.4/AIMs/AudioVisualSceneDescription.json

7.1.26.6 Reference Software

7.1.26.6.1 Disclaimers

1. This OSD-AVS Reference Software Implementation is released with the BSD-3-Clause licence.
2. The purpose of this OSD-AVS Reference Software is to show a working Implementation of OSD-AVS, 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.

7.1.26.6.2 Guide to the OSD-AVS code

OSD-AVS arranges the aligned visual and speech objects into Audio-Visual Scene Descriptors.
Use of this Reference Software for the OSD-AVS AI Module is for developers who are familiar with Python, Docker, and RabbitMQ.
The OSD-AVS 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.

7.1.26.6.3 Acknowledgements

This OSD-AVS Reference Software has been developed by the MPAI *AI Framework* Development Committee (AIF-DC).

7.1.26.7 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-AVS AIM. AIM. Conformance Testing of the individual AIMs of the OSD-AVS 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 OSD-AVS AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Speech Objects	Shall validate against Speech Objects schema. Speech Data shall conform with Qualifier.
	Audio Objects	Shall validate against Audio Objects schema. Audio Data shall conform with Qualifier.
	Visual Objects	Shall validate against Visual Objects schema. Visual Data shall conform with Qualifier.
Produces	Audio-Visual Scene Descriptors	Shall validate against AV Scene Descriptors schema.

7.1.27 Speech Scene Description

7.1.27.1 Functions

The Speech Scene Description (OSD-SSD) AIM receives Speech Objects their Space-Time information as inputs and produces the Descriptors of a Scene that is composed of Speech Objects and Speech Scenes as output. The OSD-SSD AIM may also receive an Alert conveying information on potential anomalies in the input Speech Objects:

Receives	Space-Time	of the input Objects having the same time base.
	Speech Objects	Individual Speech Objects.
	Scene Descriptors	Scene the Objects belong to.
Integrates	Space-Time and Speech Object	with Scene Descriptors.
Produces	Speech Scene Descriptors	Output#1 of AIM
	Alert	Output#2 of AIM signaling potential anomalies in Object.

7.1.27.2 Reference Model

The Reference Architecture is depicted in Figure 1.

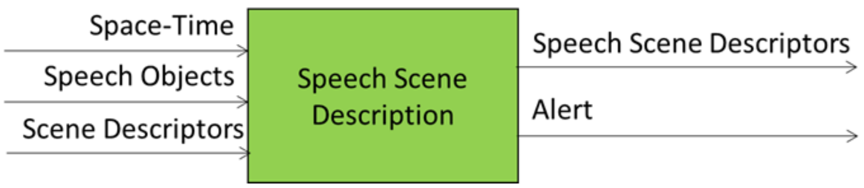

Figure 1 – The Speech Scene Description (OSD-SSD) AIM

7.1.27.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Speech Scene Description (OSD-SSD) AIM. .

Table 1 – I/O Data of the Speech Scene Description (OSD-SSD) AIM

Input	Description
[Space-Time]()	Space-Time of input Objects.
[Speech Objects]()	Input Speech Objects.
[Scene Descriptors]()	Input Scene Descriptors.
Output	**Description**
[Speech Scene Descriptors]()	The output Speech Scene Descriptors.
[Alert]()	Data signalling potential anomalies in Object.

7.1.27.4 *JSON Metadata*

https://schemas.mpai.community/OSD/V1.4/AIMs/SpeechSceneDescription.json

7.1.27.5 *Conformance Testing*

Table 2 provides the Conformance Testing Method for OSD-SSD 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 OSD-SSD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Speech Objects	Shall validate against Speech Object schema. Media-specific Data shall conform with their Qualifiers.
	Scene Descriptors	Shall validate against Scene Descriptors schema.
Produces	Speech Scene Descriptors	Shall validate against Speech Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.1.28 Visual Direction Identification

7.1.28.1 *Functions*

The Visual Direction Identification (OSD-VDI) AIM identifies the Point of View signaled by the index finger or a human body in a space described by a Visual Scene Geometry:

Receives	Visual Scene Geometry	The Geometry of the Visual Scene
	Body Descriptors	The Descriptors of a Body.
Produces	Point of View	The direction of a line traversing a point of the forefinger of the Entity.

7.1.28.2 *Reference Model*

Figure 1 depicts the Reference Model of the Visual Direction Identification (OSD-VOI) AIM.

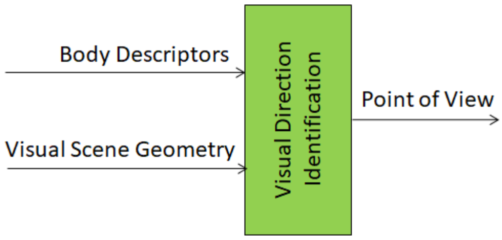

Figure 1 – The Visual Direction Identification (OSD-VOI) AIM Reference Model

7.1.28.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Visual Direction Identification (OSD-VOI) AIM.

Table 1 – I/O Data of the Visual Direction Identification (OSD-VOI) AIM

Input	Description
Body Descriptors Object	The Descriptors of the Body Objects in the Visual Scene.
Visual Scene Geometry	The digital representation of the spatial arrangement of the Visual Objects of the Scene.
Output	**Description**
Point ov View	The direction of the line traversing the forefinger of the target Entity.

7.1.28.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.4/AIMs/VisualDirectionIdentification.json

7.1.28.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-VDI 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 OSD-VDI AIM

Receives	Visual Scene Geometry	Shall validate against Visual Scene Geometry schema
	Body Descriptors Object	Shall validate against Body Descriptors XML schema
Produces	Point of View	Shall validate against Point of View schema

7.1.29 Visual Instance Identification

7.1.29.1 Functions

The Visual Instance Identification (OSD-VII) AIM receives a Visual Object provides the identifier of the Visual Object based on a Taxonomy:

Receives	*Visual Object*	To be identified.
Produces	An *Instance ID*	Identifying an element of a set of Visual Objects belonging to a level in a taxonomy.

7.1.29.2 Reference Model

Figure 1 specifies the Reference Model of the Visual Instance Identification (OSD-VII) AIM.

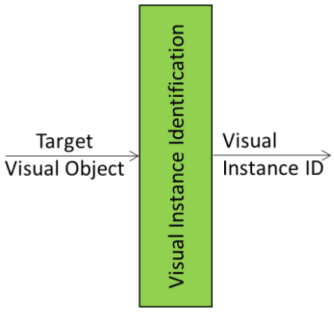

Figure 1 – The Visual Instance Identification (OSD-VII) AIM Reference Model

7.1.29.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Visual Instance Identification (OSD-VII) AIM.

Table 1 – I/O Data of Visual Instance Identification (OSD-VII) AIM

Input	**Description**
Target Visual Object	The Visual Object crossed by the line traversing the forefinger of the Entity.
Output	**Description**
Visual Instance Identifier	The Identifier of the specific Visual Object belonging to a level in the taxonomy.

7.1.29.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.4/AIMs/VisualInstanceIdentification.json

7.1.29.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-VII 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 OSD-VII AIM

Receives	Visual Object	Shall validate against Visual Object schema. Visual Data shall conform with Qualifier.
Produces	Instance ID	Shall validate against Instance ID schema.

Table 3 provides an example of MMC-AQM AIM conformance testing.

Table 3 – An example MMC-AQM AIM conformance testing

Input Data	**Data Type**	**Input Conformance Testing Data**
Input Image	JPEG	All input Text files to be drawn from Image files.
Output Data	**Data Type**	**Data Format**
Object Instance ID	Identifier	All Identifiers of Visual Objects shall be represented according to Instance Identifier

7.1.29.6 Performance Assessment

Performance Assessment of an OSD-VII AIM Implementation shall be performed using a dataset of object of a category of objects of an identified Taxonomy.

The Performance Assessment Report of an OSD-VII AIM Implementation shall include:

1. The Identifier of the OSD-VII AIM.
2. The Identifier of the object dataset.
3. The data type of object: analogue, digital, 2D, 3D etc.
4. The Performance of the OSD-VII AIM expressed as the Accuracy of the Identifiers provided by the OSD-VII AIM computed on all objects of the dataset referenced in 2.

7.1.30 Visual Object Extraction

7.1.30.1 Functions

The Visual Object Extraction (OSD-VOE) AIMs receives a Visual Scene Geometry with its Visual Objects and a Point of View and provides the Visual Object that is crossed by the Point of View:

Receives	*Visual Scene Geometry*	Spatial description of object arrangement.
	Visual Objects	To be extracted for identification.
	Point of View	Crossed by line.
Extracts	*Visual Object*	Crossed by line from Point of View.

7.1.30.2 Reference Model

Figure 1 depicts the Reference Model of the Visual Object Extraction (OSD-VOE) AIM.

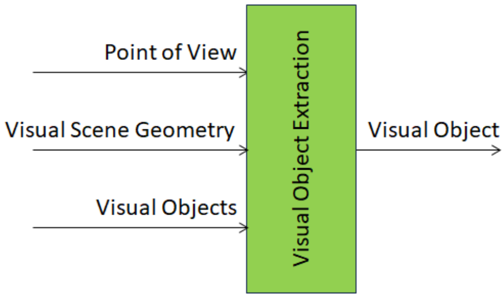

Figure 1 – The Visual Object Extraction (OSD-VOE) AIM Reference Model

7.1.30.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Visual Object Extraction (OSD-VOE) AIM.

Table 1 – I/O Data of the Visual Object Extraction (OSD-VOE) AIM

Input	Description
Point of View	The direction of the line traversing the forefinger of the Entity.

Visual Scene Geometry	The digital representation of the spatial arrangement of the Visual Objects of the Scene.
Visual Objects	The Visual Objects of the Visual Scene Geometry.
Output	**Description**
Target Visual Object Identifier	The Visual Object crossed by the line traversing the forefinger of the Entity.

7.1.30.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.4/AIMs/VisualObjectExtraction.json

7.1.30.5 Conformance Testing

Table 2 provides the Conformance Testing Method for MMC-VOE 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 MMC-VOE AIM

Receives	Visual Scene Geometry	Shall validate against Visual Scene Geometry schema.
	Visual Objects	Shall validate against Visual Object schema. Visual Data shall conform with Qualifier.
	Point of View	Shall validate against Point of View schema.
Extracts	Visual Object	Shall validate against Visual Object schema. Visual Data shall conform with Qualifier.

7.1.31 Performance Assessment

7.1.32 Visual Object Identification

7.1.32.1 Functions

The Visual Object Identification (OSD-VOI) AIM produces the Identifier based on a Taxonomy of a Visual Object included in a Visual Scene Geometry that is crossed by a Point of View:

Receives	*Visual Scene Geometry*	The arrangement of the objects in the Scene, a subset of Visual Scene Descriptors.
	Visual Objects	The Objects in the Scene.
	Body Descriptors	Descriptors of the Body indicating the object.
Produces	*Visual Instance ID*	Identifying a Visual Object in the Scene that belongs to some level in a taxonomy.

7.1.32.2 Reference Model

Figure 1 specifies the Reference Model of Visual Object Identification (OSD-VOI) AIM.

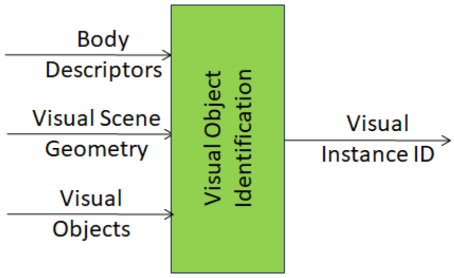

Figure - The Visual Object Identification (OSD-VOI) AIM Reference Model

7.1.32.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Visual Object Identification (OSD-VOI) AIM.

Table 1 – I/O Data of the Visual Object Identification (OSD-VOI) AIM

Input	Description
Body Descriptors Object	The Descriptors of the Body Objects of Entities in the Visual Scene.
Visual Scene Geometry	The digital representation of the spatial arrangement of the Visual Objects of the Scene.
Visual Object	The Visual Objects in the Visual Scene that are not Entities.
Output	**Description**
Visual Instance Identifier	The Identifier of the specific Visual Object belonging to a level in the taxonomy.

7.1.32.4 SubAIMs

Visual Object Identification (OSD-VOI) is a Composite AIM specified by Figure 2.

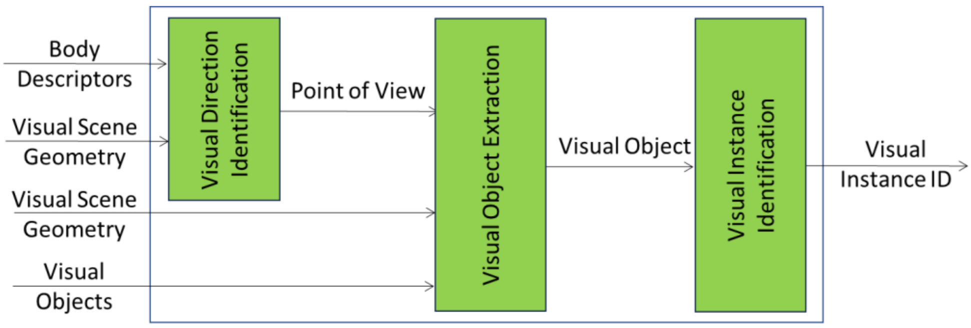

Figure 2 - The Visual Object Identification (OSD-VOI) Composite AIM

Note that the Visual Direction Identification AIM can parse either an AV Scene Geometry or its Visual Scene Geometry subset.

The AIMs composing the Visual Object Identification (OSD-VOI) Composite AIM are:

AIM	AIMs	Names	JSON
OSD-VOI		Visual Object Identification	Link

	OSD-VDI	Visual Direction Identification	Link
	OSD-VOE	Visual Object Extraction	Link
	OSD-VII	Visual Instance Identification	Link

7.1.32.5 JSON Metadata

https://schemas.mpai.community/OSD/V1.4/AIMs/VisualObjectIdentification.json

7.1.32.6 Conformance Testing

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

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 OSD-VOI AIM

Receives	Visual Scene Geometry	Shall validate against Visual Scene Geometry schema.
	Visual Objects	Shall validate against Visual Objects schema. Visual Data shall conform with Qualifier.
	Body Descriptors Object	Shall validate against Body Descriptors XML schema.
Produces	Visual Instance ID	Shall validate against Instance ID schema.

7.1.33 Visual Scene Description

7.1.33.1 Functions

The Visual Scene Description (OSD-VSD) AIM receives Visual Objects, the Descriptors of the Scene the Objects belong to, and their Space-Time information as inputs and produces the Descriptors of a Scene that is composed of Visual Objects and Scenes as output. The OSD-VSD AIM may also receive an Alert conveying information on potential anomalies in the input Visual Objects:

Receives	Space-Time	of the input Objects having the same time base.
	Visual Objects	Individual Visual Objects.
	Scene Descriptors	Scene the Objects belong to.
Integrates	Space-Time and Visual Object	with Scene Descriptors.
Produces	Visual Scene Descriptors	Output#1 of AIM
	Alert	Output#2 of AIM signalling potential anomalies in Object.

7.1.33.2 Reference Model

The Reference Architecture is depicted in Figure 1.

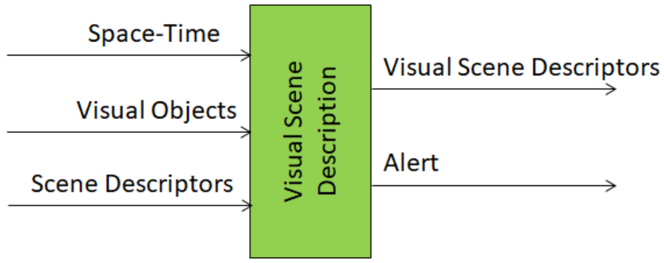

Figure 1 – The Visual Scene Description (OSD-VSD) AIM

7.1.33.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Visual Scene Description (OSD-VSD) AIM. .

Table 1 – I/O Data of the Visual Scene Description (OSD-VSD) AIM

Input	**Description**
Space-Time	Space-Time of input Objects.
Visual Objects	Input Visual Objects.
Scene Descriptors	Input Scene Descriptors.
Output	**Description**
Visual Scene Descriptors	The output Visual Scene Descriptors.
Alert	Data signalling potential anomalies in Object.

7.1.33.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.4/AIMs/VisualSceneDescription.json

7.1.33.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-VSD 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 OSD-VSD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Visual Objects	Shall validate against Visual Object schema. Media-specific Data shall conform with their Qualifiers.
	Scene Descriptors	Shall validate against Scene Descriptors schema.
Produces	Visual Scene Descriptors	Shall validate against Visual Scene Descriptors schema.

	Alert	Shall validate against Alert schema.

7.1.34 Audio-Visual Scene Rendering

7.1.34.1 Functions

The Audio-Visual Scene Rendering (PAF-AVR) AIM

1. Receives an input Point of View and all or some of the following input data: an input Portable Avatar, input Audio-Visual Scene Descriptors, and an input Spatial Attitude.
2. Produces the Audio, Speech, and Visual components resulting from the rendering from the input Point of View of one of:
    1. The input Audio-Visual Scene Descriptors - if no input Portable Avatar is present.
    2. A speaking avatar constructed according to the data of the input Portable Avatar embedded in the input Potable Avatar's Audio-Visual Scene Descriptors with the input Spatial Attitude - if the input Audio-Visual Scene Descriptors are not present.
    3. The input Audio-Visual Scene Descriptors that include an avatar constructed according to the data of the input Portable Avatar and embedded in the input Audio-Visual Scene Descriptors with the input Spatial Attitude - if both input Portable Avatar and input Audio-Visual Scene Descriptors are present.

Receives	Portable Avatar	Jointly with or alternatively with AV Scene Descriptors.
	Audio-Visual Scene Descriptors	Alternative to or superseding that of the Portable Avatar.
	Spatial Attitude	Spatial Attitude of the Avatar in the Audio-Visual Scene.
	Point of View	To be used in rendering the scene and its objects.
Transforms	Portable Avatar	Into generic Audio-Visual Scene Descriptors if input Portable Avatar is present.
Produces	Portable Avatar's Output Speech	Always integrated in the Audio-Visual Scene. Output Speech results from the rendering of Audio Scene Descriptors from human-selected Point of View.
	Output Audio	Resulting 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 also be integrated.

7.1.34.2 Reference Model

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

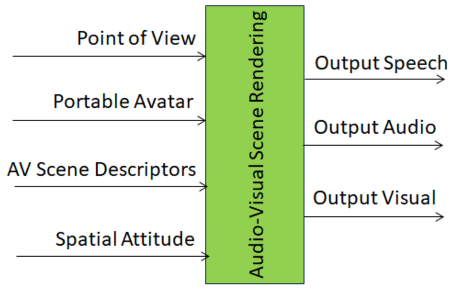

Figure 1 – The Audio-Visual Scene Rendering (PAF-AVR) AIM

7.1.34.3 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
Spatial Attitude	of the Avatar in 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 Visual Object	The Visual components of the Audio-Visual Scene.

7.1.34.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.5/AIMs/AudioVisualSceneRendering.json

7.1.34.5 Profiles

The Profiles of Audio-Visual Scene Rendering are specified.

7.1.34.6 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.
	Spatial Attitude	Shall validate against Spatial Attitude Schema.
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.

7.1.35 Face Identity Recognition

7.1.35.1 Functions

The Face Identity Recognition (PAF-FIR) AIM receives an input Visual Object representing a Face and produces a Bounding Box with the Face and the Identifier of the Face. The PAF-FIR AIM may also receive an input Visual Geometry of the Visual Scene, an input Time, and a Text Object related to the containing the input Visual Object :

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 Geometry	Of the scene where the Face is located.
Searches for	*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.

7.1.35.2 Reference Model

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

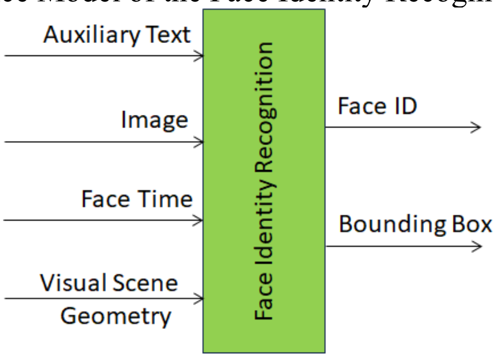

Figure 1 – Face Identity Recognition AIM

7.1.35.3 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 Text Object	Text with a content related to Face ID.
Image Visual Object	An image containing the Face to be identified.
Face Time	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.

7.1.35.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.5/AIMs/FaceIdentityRecognition.json

7.1.35.5 Reference Software

7.1.35.5.1 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.

7.1.35.5.2 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

7.1.35.5.3 Acknowledgements

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

7.1.35.6 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 Time	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.

	Bounding Boxes	Shall validate against Bounding Box Schema. Bounding Box Data shall conform with Visual Qualifier.

7.1.35.7 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.

7.1.36 Entity Body Description

7.1.36.1 Functions

The Entity Body Description (PAF-EBD) AIM receives an input Visual Object representing a Body and produces the Descriptors of the input Visual Object:

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

7.1.36.2 Reference Model

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

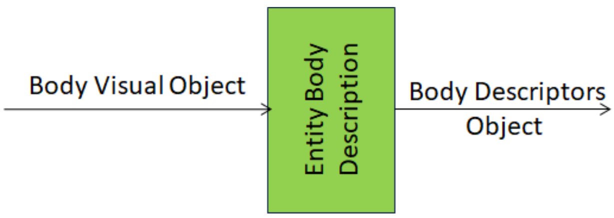

Figure 1 Entity Body Description (PAF-EBD) AIM

7.1.36.3 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

Input	**Description**
Body Visual Object	Visual Object representing the body of an Entity.
Output	**Description**
Body Descriptors Object	Body Descriptors of Visual Object.

7.1.36.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.5/AIMs/EntityBodyDescription.json

7.1.36.5 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 Object	Shall validate against Visual Object Schema. Body Data shall conform with Visual Object.
Produces	Body Descriptors Object	Shall validate against Body Descriptors XML Schema.

7.1.37 Entity Face Description

7.1.37.1 Functions

The Entity Face Description (PAF-EFD) AIM receives an input Visual Object representing a Face and produces the Descriptors of the input Visual Object:

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

7.1.37.2 Reference Model

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

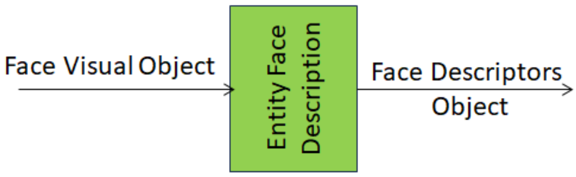

Figure 1 - Entity Face Description (PAF-EFD) AIM

7.1.37.3 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	Entity Face to be Described.
Output	**Description**
Face Descriptors Object	Descriptors of Face.

7.1.37.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.5/AIMs/EntityFaceDescription.json

7.1.37.5 Reference Software

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

7.1.37.6 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 Object	Shall validate against Face Descriptors Schema.

7.1.37.7 Performance Assessment

7.1.38 Portable Avatar Demultiplexing

7.1.38.1 Functions

The Portable Avatar Demultiplexing (PAF-PDX) AIM extracts the components of an input Portable Avatar - Portable Avatar ID, Avatar Space-Time, Avatar, Language Selector, Speech Object, Text Object, Speech Model, Personal Status, Audio-Visual Scene Descriptors, and Audio-Visual Scene Space Time:

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

7.1.38.2 Reference Model

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

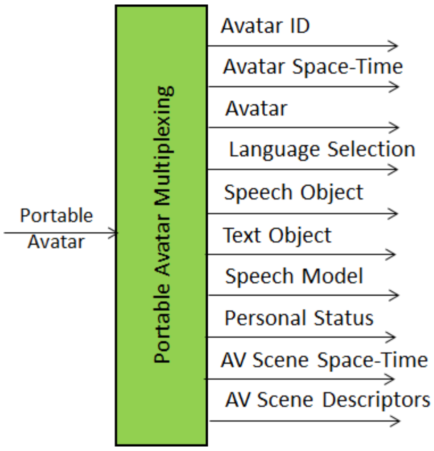

Figure 1– The Personal Avatar Demultiplexing (PAF-PDX) AIM

7.1.38.3 Input/Output Data

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

Table 1 – 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 Model	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.

7.1.38.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.5/AIMs/PortableAvatarDemultiplexing.json

7.1.38.5 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 Selector	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.

7.1.39 PS-Face Interpretation

7.1.39.1 Functions

The PS-Face Interpretation (PAF-PFI) AIM receives input Faces Descriptors and produces the Face Personal Status from the input Face Descriptors:

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

7.1.39.2 Reference Model

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

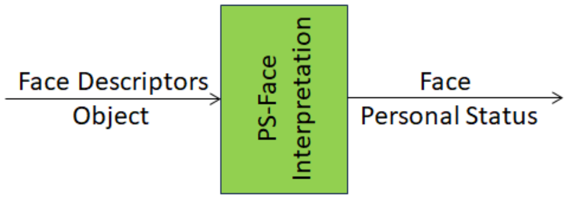

Figure 1– The PS-Face Interpretation (PAF-PFI) AIM Reference Model

7.1.39.3 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

Input	Description
Face Descriptors Object	Descriptors of Face
Output	**Description**
Face Personal Status	Personal Status of Face

7.1.39.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.5/AIMs/PSFaceInterpretation.json

7.1.39.5 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 Descriptors Object	Shall validate against Face Descriptors Object Schema
Produces	Face Personal Status	Shall validate against Face Personal Status Schema

7.1.40 PS-Gesture Interpretation

7.1.40.1 Functions

The PS-Gesture Interpretation (PAF-PGI) AIM receives input Gesture Descriptors and produces the Gesture Personal Status from the input Gesture Descriptors:

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

7.1.40.2 Reference Model

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

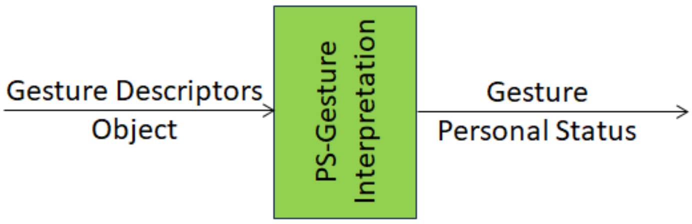

Figure 1– The PS-Gesture Interpretation (PAF-PGI) AIM Reference Model

7.1.40.3 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 Object	Descriptors of Gesture
Output	**Description**
Gesture Personal Status	Personal Status of Gesture

7.1.40.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.5/AIMs/PSGestureInterpretation.json

7.1.40.5 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 Descriptors Object	Shall validate against Gesture Descriptors Schema
Produces	Gesture Personal Status	Shall validate against Gesture Personal Status Schema

7.1.41 Personal Status Display

7.1.41.1 Functions

The Personal Status Display (PAF-PSD) AIM receives an input Identifier of a conversational machine, an input Avatar Model, and an input Text Object and produces a Portable Avatar that includes the input Identifier, an Avatar uttering a Speech Object synthesised from the input Text Object and based on the input Avatar Model. The PAF-PSD AIM may also receive an input Personal Status and Speech Model that can use to synthesise the Speech Object from the input Text Object and produce a speaking Avatar that displays the input Personal Status:

Receives	Machine ID	ID to be used to identify the Avatar in Portable Avatar.
	Text Object	Text associated to Avatar in Portable Avatar.
	Personal Status	Personal Status associated to Avatar in Portable Avatar.
	Avatar Model	3D Model associated to Avatar in Portable Avatar.
	Speech Model	Speech Model Associated to Avatar in Portable Avatar.
Produces	Portable Avatar	Output Portable Avatar.
Enables	PAF-AVR	To render the Portable Avatar produced by PAF-PSD.

7.1.41.2 Reference Model

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

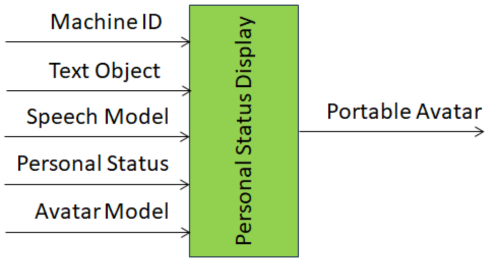

Figure 1 – Reference Model of Personal Status Display (PAF-PSD) AIM

7.1.41.3 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	**Description**
Avatar ID	Portable Avatar's ID from Upstream AIM

Avatar Model	Part of Portable Avatar from Upstream AIM or embedded in PSD
Text Object	Texts of Portable Avatar from Keyboard or upstream AIM
Personal Status	To add PS to Speech, Face, and Gesture from Personal Status Extraction or Machine
Speech Model	Neural Network from Upstream AIM or embedded in PSD
Output data	**Description**
Portable Avatar	Output Portable Avatar to downstream AIM or renderer.

7.1.41.4 SubAIMs

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

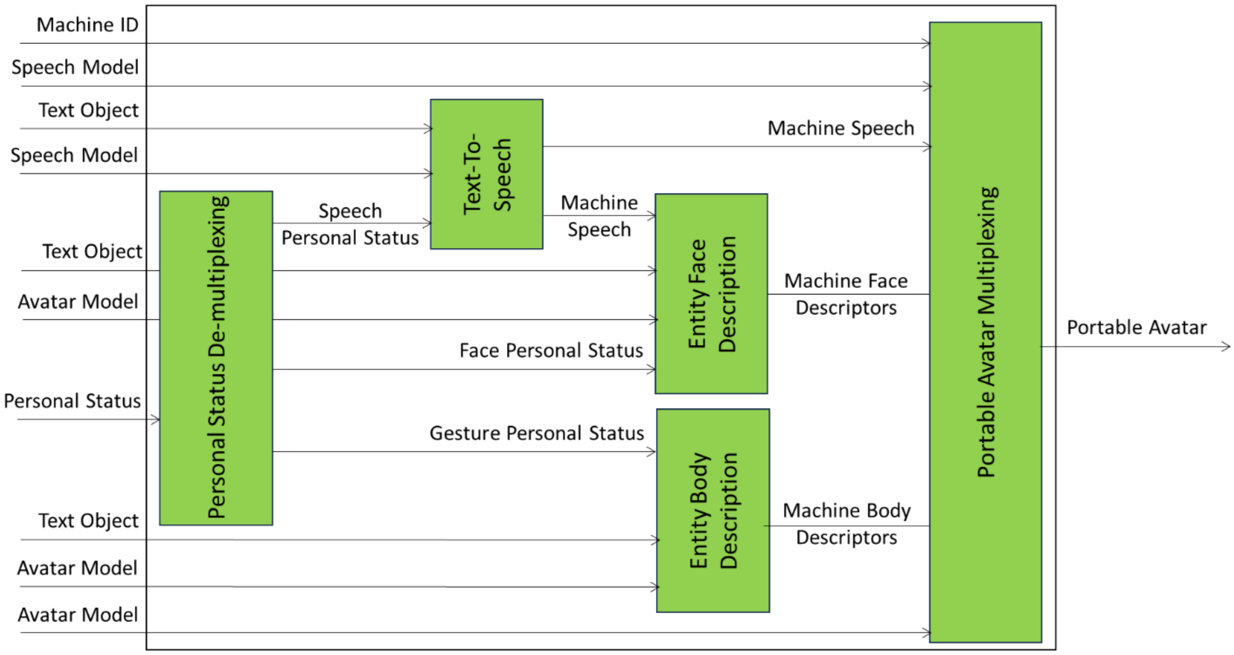

Figure 2 – Reference Model of Personal Status Display Composite AIM

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 Entity Face Descriptors.
5. PS-Gesture and Text are used for Entity 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.

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

AIW	**AIMs**	**Name and Specification**	**JSON**
PAF-PSD		Personal Status Display	X
	MMC-PDX	Personal Status Demultiplexing	X

	MMC-TTS	Text-to-Speech	X
	PAF-EFD	Entity Face Description	X
	PAF-EBD	Entity Body Description	X
	PAF-PMX	Portable Avatar Multiplexing	X

7.1.41.5 JSON Metadata

https://schemas.mpai.community/PAF/V1.5/AIMs/PersonalStatusDisplay.json

7.1.41.6 Profiles

The Profiles of Personal Status Display are specified.

7.1.41.7 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 Model	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.

7.1.42 Portable Avatar Multiplexing

7.1.42.1 Functions

The Portable Avatar Multiplexing (PSD-PMX) AIM Multiplexes the input component elements of a Portable Avatar - Portable Avatar ID, Avatar Space-Time, Avatar, Language Selector, Speech Object, Text Object, Speech Model, Personal Status, Audio-Visual Scene Descriptors, and Audio-Visual Scene Space Time into 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

7.1.42.2 Reference Model

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

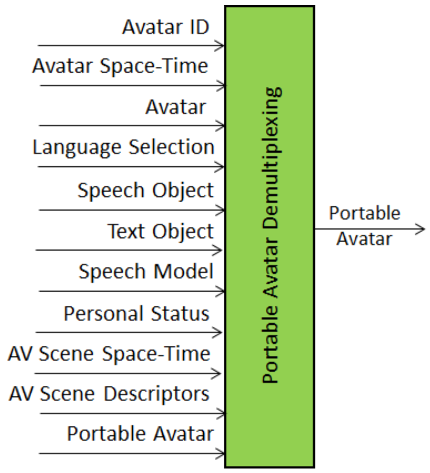

Figure 1 – The Portable Avatar Multiplexing (PSD-PMX) AIM

7.1.42.3 Input/Output Data

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

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

Input	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 Model	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.

7.1.42.4 JSON Metadata

https://schemas.mpai.community/PAF/V1.5/AIMs/PortableAvatarMultiplexing.json

7.1.42.5 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	AvatarID	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 Selector	Shall validate against Selector Schema
	Speech Object	Shall validate against Speech Object Schema. Text Data shall conform with SpeechQualifier.
	Text Object	Shall validate against Text Object Schema. Speech Data shall conform with Text Qualifier.
	Speech Model	Shall validate against Machine Learning Model Schema. Speech Model Data shall conform with Machine Learning Model Qualifier.
	Avatar Personal Status	Shall validate against Personal Status Schema.
	AV Scene Descriptors	Shall validate against AV Scene Descriptors Schema.

	AV 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.

7.2 Reference Software

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

1. The purpose of the Reference Software is to provide a working Implementation of an AIM, not a ready-to-use product.
2. MPAI disclaims the suitability of the Reference Software for any other purposes than those of the MPAI-HMC 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 the Reference Software, e.g., by accepting the licences from third-party repositories.

Note that at this stage only part of the MPAI-HMC AIMs have a Reference Software Implementation.

7.3 Conformance Testing

An implementation of an AI Module conforms with MPAI-HMC if it accepts as input and produces as output Data and/or Data Objects (combination of Data of a certain Data Type and its Qualifier) conforming with those specified by all relevant MPAI Technical Specifications.
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 has the specified type.
- 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.

7.4 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 AIM 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., the word error rate (WER) of an Automatic Speech Recognition (ASR) AIM.
2. *Bias*: Performance Assessment measures how well an AIM performs its function, using a metric that depends on a bias related to certain attributes of the AIM. For instance, an ASR AIM tends to have a higher WER when the speaker is from a particular geographic area.
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.

Note that the current MPAI-HMC V2.1 Technical Specification provides AIM Performance Assessment methods for a limited number of AIMs.

8 Data Types

MPAI-HMC V2.1 only uses Data Types defined by other MPAI Technical Specifications. Table 1 provides the full list with web links of the Data Types utilised by HMC-CEC organised according to the Technical Specifications that specify them.

Table 1 - Data Types utilised by HMC-CEC

MPAI-AIF	MPAI-OSD	MPAI-OSD
Machine Learning Model	3D Model Object	Speech Object
MPAI-MMC	Audio Object	Speech Scene Descriptors
Cognitive State	Audio Scene Descriptors	Speech Scene Geometry
Emotion	Audio Scene Geometry	Text Object
Intention	Basic Audio-Visual Scene Descriptors	Time
Meaning	Basic Audio-Visual Scene Geometry	Basic Visual Scene Descriptors
Personal Status	Audio-Visual Event Descriptors	Basic Visual Scene Geometry
Social Attitude	Audio-Visual Object	Visual Object
Speech Descriptors	Audio-Visual Scene Descriptors	Visual Scene Descriptors
Text Descriptors	Audio-Visual Scene Geometry	Visual Scene Geometry
	Instance Identifier	**MPAI-PAF**
	Point of View	Avatar
	Selector	Body Descriptors Object
	Space-Time	Face Descriptors Object
	Spatial Attitude	Portable Avatar

8.1 Machine Learning Model

8.1.1 Definition

A Data Type an instance of which results from the application of training data to a process.

8.1.2 Functional Requirements

A Machine Learning Model enables the performance of specific functions such as classification most of which are target of MPAI Technical Specifications.

8.1.3 Syntax

https://schemas.mpai.community/AIF/V2.1/data/MLModel.json

8.1.4 Semantics

Label	Size	Description
Header	N1 Bytes	Machine Learning Model Header
- Standard-MachineLearningModel	9 Bytes	The characters “AIF-MLM-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
MLModelID	N5 Bytes	Identifier of Machine Learning Model.
MLModelQualifier	N6 Bytes	Qualifier of Machine Learning Model
MLModelDataLength	N7 Bytes	Length in Bytes of ML Model, i.e., Base Image that is required to operate the ML Model.
MLModeDataURI	N8 Bytes	URI of Machine Learning Model Data
DescrMetadata	N9 Bytes	Descriptive Metadata.

8.2 Cognitive State

8.2.1 Definition

Cognitive State is a Personal Status Factor representing the internal state of an Entity such as “surprised” or “interested”.

8.2.2 Functional Requirements

Cognitive State can be expressed via several *Modalities*: Text, Speech, Face, and Gestures. (Other Modalities, such as body posture, may be handled in future MPAI Versions.)
Within a given Modality, Cognitive State can be analysed and interpreted via various *Descriptors*. For example, when expressed via Speech, the elements may be expressed through combinations of such features as prosody (pitch, rhythm, and volume variations); separable speech effects (such as degrees of voice tension, breathiness, etc.); and vocal gestures (laughs, sobs, etc.).
Cognitive State is represented by a standard set of labels and associated semantics by two tables:

- A *Label Set Table* containing descriptive labels relevant to the Factor in a three-level format:
  - The CATEGORIES column specifies the relevant categories using nouns (e.g., “ANGER”).
  - The GENERAL ADJECTIVAL column gives adjectival labels for general or basic labels within a category (e.g., “angry”).
  - The SPECIFIC ADJECTIVAL column gives more specific (sub-categorised) labels in the relevant category (e.g., “furious”).
- A *Label Semantics Table* providing the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns of the Label Set Table. For example, for “angry” the semantic gloss is “emotion due to perception of physical or emotional damage or threat.”

Table 1 gives the standardised three-level Basic Cognitive State Label Set.

Table 1 – Basic Cognitive State Label Set

EMOTION CATEGORIES	**GENERAL ADJECTIVAL**	**SPECIFIC ADJECTIVAL**
AROUSAL	aroused/excited/energetic	cheerful
		playful
		lethargic
		sleepy
ATTENTION	attentive	expectant/anticipating
		thoughtful
		distracted/absent-minded
		vigilant
		hopeful/optimistic
BELIEF	credulous	
	skeptical	
INTEREST	interested	fascinated
		curious
		bored
SURPRISE	surprised	astounded
		startled
UNDERSTANDING	comprehending	uncomprehending
		bewildered/puzzled

EMOTION CATEGORIES	**GENERAL ADJECTIVAL**	**SPECIFIC ADJECTIVAL**
AROUSAL	aroused/excited/energetic	cheerful
		playful
		lethargic
		sleepy
ATTENTION	attentive	expectant/anticipating
		thoughtful
		distracted/absent-minded
		vigilant
		hopeful/optimistic
BELIEF	credulous	
	skeptical	
INTEREST	interested	fascinated
		curious

		bored
SURPRISE	surprised	astounded
		startled
UNDERSTANDING	comprehending	uncomprehending
		bewildered/puzzled

Table 2 provides the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns above.

Table 2 – Basic Cognitive State Semantics Set

ID	Cognitive State	Meaning
1	aroused/excited/energetic	cognitive state of alertness and energy
2	astounded	high degree of surprised
3	attentive	cognitive state of paying attention
4	bewildered/puzzled	high degree of incomprehension
5	bored	not interested
6	cheerful	energetic combined with and communicating happiness
7	comprehending	cognitive state of successful application of mental models to a situation
8	credulous	cognitive state of conformance to mental models of a situation
9	curious	interest due to drive to know or understand
10	distracted/absent-minded	not attentive to present situation due to competing thoughts
11	expectant/anticipating	attentive to (expecting) future event or events
12	fascinated	high degree of interest
13	interested	cognitive state of attentiveness due to salience or appeal to emotions or drives
14	lethargic	not aroused
15	playful	energetic and communicating willingness to play
16	sceptical	not credulous
17	sleepy	not aroused due to need for sleep
18	surprised	cognitive state due to violation of expectation
19	startled	surprised by a sudden event or perception
20	surprised	cognitive state due to violation of expectation
21	thoughtful	attentive to thoughts
22	uncomprehending	not comprehending

These sets have been compiled in the interests of basic cooperation and coordination among AIM submitters and vendors complemented by a procedure whereby AIM submitters may propose extended or alternate sets for their purposes.

An Implementer wishing to extend or replace a *Label Set Table* for one of the three Factors is requested to do the following:

1. Create a new Label Set Table where:
    1. Proposed additions are clearly marked (in case of extension).
    2. b. All the elements of the target Cognitive State and levels (up to 3) are listed (in case of replacement).
2. Create a new Label Semantics Table where the semantics of elements of the Cognitive State is:
    1. Added to the semantics of the existing Cognitive State (in case of extension).
    2. Provided (in case of replacement).
       The submitted semantics should have a level of detail comparable to the semantics given in the current *Label Semantics Table*.
3. Submit both tables to the MPAI Secretariat.

The appropriate MPAI Development Committee will examine the proposed extension or replacement. Only the adequacy of the proposed new tables in terms of clarity and completeness will be considered. In case the new tables are not clear or complete, a revision of the tables will be requested.

The accepted Cognitive State Set will be identified as proposed by the submitter and reviewed by the appropriate MPAI Committee and posted to the MPAI web site.

The versioning system is based on a name – MPAI for MPAI-generated versions or "organisation name" for the proposing organisation – with a suffix m.n where m indicates the version and n indicated the subversion.

8.2.3 Syntax

https://schemas.mpai.community/MMC/V2.4/data/CognitiveState.json

8.2.4 Semantics

Label	Description
Header	Entity Cognitive State Header
- Standard-EntityCognitiveState	The characters "MMC-ECS-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
EntityCognitiveStateID	Identifier of CogState.
EntityCognitiveStateSpaceTime	Space-Time info of CogState.
EntityCognitiveStateData	Data associated to CogState.
- FusedCogState	Integrated CogState Value.
- TextCogState	Text CogState Value.
- SpeechCogState	Speech CogState Value.
- FaceCogState	Face CogState Value.

- GestureCogState	Gesture CogState Value.
DescrMetadata	Descriptive Metadata

8.2.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Entity Cognitive State (MMC-ECS) if:

1. The Data validates against the Entity Cognitive State 's JSON Schema.
2. All Data in the Entity Cognitive State 's JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers if present.

8.3 Emotion

8.3.1 Definition

Emotion is a Personal Status Factor representing the internal state of an Entity such as that resulting from its interaction with the Context, such as "Angry", "Sad", "Determined".

8.3.2 Functional Requirements

Emotion can be expressed via several *Modalities*: Text, Speech, Face, and Gestures. (Other Modalities, such as body posture, may be handled in future MPAI Versions.)

Within a given Modality, Emotion can be analysed and interpreted via various *Descriptors*. For example, when expressed via Speech, the elements may be expressed through combinations of such features as prosody (pitch, rhythm, and volume variations); separable speech effects (such as degrees of voice tension, breathiness, etc.); and vocal gestures (laughs, sobs, etc.).

Emotion is represented by a standard set of labels and associated semantics by two tables:

- A *Label Set Table* containing descriptive labels relevant to the Factor in a three-level format:
    - The CATEGORIES column specifies the relevant categories using nouns (e.g., "ANGER").
    - The GENERAL ADJECTIVAL column gives adjectival labels for general or basic labels within a category (e.g., "angry").
    - The SPECIFIC ADJECTIVAL column gives more specific (sub-categorised) labels in the relevant category (e.g., "furious").
- A *Label Semantics Table* providing the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns of the Label Set Table. For example, for "angry" the semantic gloss is "emotion due to perception of physical or emotional damage or threat."

Table 1 gives the standardised three-level Basic Emotion Set partly based on Paul Eckman [19].

Table 1 – Basic Emotion Label Set

EMOTION CATEGORIES	**GENERAL ADJECTIVAL**	**SPECIFIC ADJECTIVAL**
ANGER	angry	furious
		irritated
		frustrated
CALMNESS	calm	peaceful/serene

		resigned
DISGUST	disgusted	repulsed
FEAR	fearful/scared	terrified
		anxious/uneasy
HAPPINESS	happy	joyful
		content
		delighted
		amused
HURT	hurt	insulted/offended
		resentful/disgruntled
		bitter
	jealous	
PRIDE/SHAME	proud	
	ashamed	guilty/remorseful/sorry
		embarrassed
RETROSPECTION	nostalgic	homesick
SADNESS	sad	lonely
		grief-stricken
		depressed/gloomy
		disappointed

Table 2 provides the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns above.

Table 2 – Basic Emotion Semantics Set

ID	Emotion	Meaning
1	amused	positive emotion combined with interest (cognitive state)
2	angry	emotion due to perception of physical or emotional damage or threat
3	anxious/uneasy	low or medium degree of fear, often continuing rather than instant
4	ashamed	emotion due to awareness of violating social or moral norms
5	bitter	persistently angry due to disappointment or perception of hurt or injury
6	calm	relatively lacking emotion
7	content	medium or low degree of happiness, continuing rather than instant
8	delighted	high degree of happiness, often combined with surprise

9	depressed/gloomy	high degree of sadness, continuing rather than instant, combined with lethargy (see AROUSAL)
10	disappointed	sadness due to failure of desired outcome
11	disgusted	emotion due to urge to avoid, often due to unpleasant perception or disapproval
12	embarrassed	shame due to consciousness of violation of social conventions
13	fearful/scared	emotion due to anticipation of physical or emotional pain or other undesired event or events
14	frustrated	angry due to failure of desired outcome
15	furious	high degree of angry
16	grief-stricken	sadness due to loss of an important social contact
17	happy	positive emotion, often continuing rather than instant
18	homesick	sad due to absence from home
19	hurt	emotion due to perception that others have caused social pain or embarrassment
20	insulted/offended	emotion due to perception that one has been improperly treated socially
21	irritated	low or medium degree of angry
22	jealous	emotion due to perception that others are more fortunate or successful
23	joyful	high degree of happiness, often due to a specific event
24	repulsed	high degree of disgusted
25	lonely	sad due to insufficient social contact
26	mortified	high degree of embarrassment
27	nostalgic	emotion associated with pleasant memories, usually of long before
28	peaceful/serene	calm combined with low degree of happiness
29	proud	emotion due to perception of positive social standing
30	resentful/disgruntled	emotion due to perception that one has been improperly treated
31	resigned	calm due to acceptance of failure of desired outcome, often combined with low degree of sadness
32	sad	negative emotion, often continuing rather than instant, often associated with a specific event
33	terrified	high degree of fear

These sets have been compiled in the interests of basic cooperation and coordination among AIM submitters and vendors complemented by a procedure whereby AIM submitters may propose extended or alternate sets for their purposes.
An Implementer wishing to extend or replace a *Label Set Table* for Emotion is requested to do the following:

1. Create a new Label Set Table where:
    1. Proposed additions are clearly marked (in case of extension).
    2. b. All the elements of the Emotion and levels (up to 3) are listed (in case of replacement).
2. Create a new Label Semantics Table where the semantics of elements of the Emotion is:

    1. Added to the semantics of the existing Emotion (in case of extension).
    2. Provided (in case of replacement).
       The submitted semantics should have a level of detail comparable to the semantics given in the current *Label Semantics Table*.
3. Submit both tables to the [MPAI Secretariat](#).

The appropriate MPAI Development Committee will examine the proposed extension or replacement. Only the adequacy of the proposed new tables in terms of clarity and completeness will be considered. In case the new tables are not clear or complete, a revision of the tables will be requested.

The accepted Emotion Set will be identified as proposed by the submitter and reviewed by the appropriate MPAI Committee and posted to the [MPAI web site](#).

The versioning system is based on a name – MPAI for MPAI-generated versions or "organisation name" for the proposing organisation – with a suffix m.n where m indicates the version and n indicated the subversion.

8.3.3 Syntax

https://schemas.mpai.community/MMC/V2.4/data/Emotion.json

8.3.4 Semantics

Label	Description
Header	Entity Emotion Header
- Standard-EntityEmotion	The characters "MMC-EEM-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
EntityEmotionID	Identifier of the Emotion.
EntityEmotionSpaceTime	Space-Time info of Emotion
EntityEmotionData	Data associated to Emotion.
- FusedEmotion	Integrated Emotion Value.
- TextEmotion	Text Emotion Value.
- SpeechEmotion	Speech Emotion Value.
- FaceEmotion	Face Emotion Value.
- GestureCogState	Gesture Emotion Value.
DescrMetadata	Descriptive Metadata

8.3.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Entity Emotion (MMC-EEM) if:
1. The Data validates against the Entity Emotion 's JSON Schema.
2. All Data in the Entity Emotion 's JSON Schema

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

8.4 Intention

8.4.1 Definition

Data Type expressing the result of analysis of the goal of a question.

8.4.2 Functional Requirements

Intention provides abstracts of Intention of User Question using properties: qtopic, qfocus, qLAT, qSAT and qdomain.

8.4.3 Syntax

https://schemas.mpai.community/MMC/V2.4/data/Intention.json

8.4.4 Semantics

Label	Description
Header	The Intention Header
- Standard-Intention	The characters "MMC-INT-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
IntentionID	ID of Intention
IntentionData	Data included in Intention.
- qtopic	Indicates the topic of the question. Question topic is the object or event that the question is about. Ex. of Qtopic is King Lear in "Who is the author of King Lear?".
- qfocus	Indicates the focus of the question, which is the part of the question that, if replaced by the answer, makes the question a stand-alone statement. Ex. What, where, who, what policy. Which river, etc. Example: - Question: Who is the president of USA? (The word "Who" is the focus of the question and it will be replaced by "Biden" in the Answer.) - Answer: Biden is the president of USA.
- qLAT	Indicates the lexical answer type of the question.
- qSAT	Indicates the semantic answer type of the question. QSAT corresponds to Named Entity type of the language analysis results.
- qdomain	Indicates the domain of the question such as "science", "weather", "history". Example: Who is the third king of Yi dynasty in Korea? (qdomain: history)
DescrMetadata	Descriptive Metadata

8.4.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Intention (OSD-INT) if:

1. The Data validates against the Intention's JSON Schema.
2. All Data in the Intention's JSON Schema have the specified type.

8.5 Meaning

8.5.1 Definition

A Data Type representing the syntactic and semantic information of an input text. Meaning is synonym of Text Descriptors.

8.5.2 Functional Requirements

Meaning is used to extract information from text to help the Entity Dialogue Processing AIM to produce a response.

8.5.3 Syntax

https://schemas.mpai.community/MMC/V2.4/data/Meaning.json

8.5.4 Semantics

Label	Description
Header	Meaning Header
- Standard-Meaning	The characters "MMC-TXD-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MeaningID	Identifier of Meaning.
Meaning	Data set of Meaning
- POS_tagging	Results of POS (Part of Speech, e.g., noun, verb, etc.) tagging including information on the question's POS tagging set and tagged results.
- NE_tagging	Results of NE (Named Entity e.g., Person, Organisation, Fruit, etc.) tagging results including information on the question's tagging set and tagged results.
- Dependency_tagging	Results of dependency (structure of the sentence, e.g., subject, object, head of relation, etc.) tagging including information on the question's dependency tagging set and tagged results.
- SRL_tagging	Results of SRL (Semantic Role Labelling) tagging results including information on the question's SRL tagging set and tagged results. SRL indicates the semantic structure of the sentence such as agent, location, patient role, etc.
DescrMetadata	Descriptive Metadata

8.5.5 Conformance Testing

A Data instance Conforms with MPAI-MMC Meaning (MMC-MEA) if:

1. The Data validates against the Meaning's JSON Schema.

2. All Data in the Meaning's JSON Schema have the specified type.

8.6 Personal Status

8.6.1 Definition

A Data Type representing the information internal to an Entity that characterises their behaviour.

8.6.2 Functional Requirements

Personal Status is a Data Type composed of three *Factors*:

1. *Emotion* (such as "angry" or "sad").
2. *Cognitive State* (such as "surprised" or "interested").
3. *Social Attitude* (such as "polite" or "arrogant").

Factors are expressed by *Modalities*: Text, Speech, Face, and Gestures. (Other Modalities, such as body posture, may be handled in future MPAI Versions.)

Within a given Modality, the Factors can be analysed and interpreted via various *Descriptors*. For example, when expressed via Speech, the elements may be expressed through combinations of such features as prosody (pitch, rhythm, and volume variations); separable speech effects (such as degrees of voice tension, breathiness, etc.); and vocal gestures (laughs, sobs, etc.).

Each of Emotion, Cognitive State, and Social Attitude Factors is represented by a standard set of labels and associated semantics. For each of these Factors, two tables are provided:

- A *Label Set Table* containing descriptive labels relevant to the Factor in a three-level format:
  - The CATEGORIES column specifies the relevant categories using nouns (e.g., "ANGER").
  - The GENERAL ADJECTIVAL column gives adjectival labels for general or basic labels within a category (e.g., "angry").
  - The SPECIFIC ADJECTIVAL column gives more specific (sub-categorised) labels in the relevant category (e.g., "furious").
- A *Label Semantics Table* providing the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns of the Label Set Table. For example, for "angry" the semantic gloss is "emotion due to perception of physical or emotional damage or threat."

These sets have been compiled in the interests of basic cooperation and coordination among AIM submitters and vendors complemented by a procedure whereby AIM submitters may propose extended or alternate sets for their purposes.

An Implementer wishing to extend or replace a *Label Set Table* for one of the three Factors is requested to do the following:

1. Create a new Label Set Table where:
   1. Proposed additions are clearly marked (in case of extension).
   2. b. All the elements of the target Factor and levels (up to 3) are listed (in case of replacement).
2. Create a new Label Semantics Table where the semantics of elements of the target Factor is:
   1. Added to the semantics of the existing target Factor (in case of extension).
   2. Provided (in case of replacement).
      The submitted semantics should have a level of detail comparable to the semantics given in the current *Label Semantics Table*.
3. Submit both tables to the MPAI Secretariat (secretariat@mpai.community).

The appropriate MPAI Development Committee will examine the proposed extension or replacement. Only the adequacy of the proposed new tables in terms of clarity and completeness will be considered. In case the new tables are not clear or complete, a revision of the tables will be requested.

The accepted External Factor Set will be identified as proposed by the submitter and reviewed by the appropriate MPAI Committee and posted to the MPAI web site.
The versioning system is based on a name – MPAI for MPAI-generated versions or "organisation name" for the proposing organisation – with a suffix m.n where m indicates the version and n indicated the subversion.

8.6.3 Syntax

https://schemas.mpai.community/MMC/V2.4/data/PersonalStatus.json

8.6.4 Semantics

Label	Description
Header	Personal Status Header
- Standard-PersonalStatus	The characters "MMC-EPS-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
PersonalStatusID	Identifier of Meaning.
PersonalStatusSpaceTime	Space-Time info of PersonalStatus
PersonalStatus	Personal Status
- CognitiveState	Cognitive State component of Personal Status
- Emotion	Emotion component of Personal Status
- SocialAttitude	Social Attitude component of Personal Status
DescrMetadata	Descriptive Metadata

8.6.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Personal Status (MMC-EPS) if:

1. The Data validates against the Personal Status's JSON Schema.
2. All Data in the Personal Status's JSON Schema
    1. Have the specified type.
    2. Validate against their JSON Schemas.
    3. Conform with their Data Qualifiers if present.

8.7 Social Attitude

8.7.1 Definition

Social Attitude is a Personal Status Factor representing the internal state of an Entity related to the way it intends to position itself vis-à-vis the Context, e.g., "Respectful", "Confrontational", "Soothing"..

8.7.2 Functional Requirements

Social Attitude can be expressed via several *Modalities*: Text, Speech, Face, and Gestures. (Other Modalities, such as body posture, may be handled in future MPAI Versions.)
Within a given Modality, Social Attitude can be analysed and interpreted via various *Descriptors*. For example, when expressed via Speech, the elements may be expressed through combinations of such features as prosody (pitch, rhythm, and volume variations); separable

speech effects (such as degrees of voice tension, breathiness, etc.); and vocal gestures (laughs, sobs, etc.).
Social Attitude is represented by a standard set of labels and associated semantics by two tables:

- A *Label Set Table* containing descriptive labels relevant to the Social Attitude in a three-level format:
    - The CATEGORIES column specifies the relevant categories using nouns (e.g., "ANGER").
    - The GENERAL ADJECTIVAL column gives adjectival labels for general or basic labels within a category (e.g., "angry").
    - The SPECIFIC ADJECTIVAL column gives more specific (sub-categorised) labels in the relevant category (e.g., "furious").
- A *Label Semantics Table* providing the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns of the Label Set Table. For example, for "angry" the semantic gloss is "emotion due to perception of physical or emotional damage or threat."

These sets have been compiled in the interests of basic cooperation and coordination among AIM submitters and vendors complemented by a procedure whereby AIM submitters may propose extended or alternate sets for their purposes.
An Implementer wishing to extend or replace a *Label Set Table* for Social Attitude is requested to do the following:

1. Create a new Label Set Table where:
    1. Proposed additions are clearly marked (in case of extension).
    2. b. All the elements of the target Social Attitude and levels (up to 3) are listed (in case of replacement).
2. Create a new Label Semantics Table where the semantics of elements of the Social Attitude is:
    1. Added to the semantics of the existing Social Attitude (in case of extension).
    2. Provided (in case of replacement).
       The submitted semantics should have a level of detail comparable to the semantics given in the current *Label Semantics Table*.
3. Submit both tables to the MPAI Secretariat.

Table 1 gives the standardised three-level Basic Social Attitude Set.

Table 1 – Basic Social Attitude Label Set

EMOTION CATEGORIES	**GENERAL ADJECTIVAL**	**SPECIFIC ADJECTIVAL**
ACCEPTANCE	accepting	friendly
		welcoming/inviting
	exclusive/cliquish	unfriendly/hostile
AGREEMENT/ DISAGREEMENT	like-minded	
	argumentative/disputatious	sarcastic
AGGRESSION	aggressive	combative/belligerent
		passive-aggressive
		mocking
	peaceful	
	submissive	
	admiring/approving	awed

APPROVAL/DISAPPROVAL		flattering
		laudatory
		congratulatory
	disapproving	contemptuous
		critical
		belittling
	indifferent	
ACTIVITY/PASSIVITY	assertive	controlling
	passive	permissive/lenient
COOPERATION	cooperative/agreeable	flexible
		supportive
		reasonable
		communicative
	uncooperative	stubborn
		disagreeable
		subversive/undermining
		uncommunicative
EMPATHY	empathetic/caring	kind
		sympathetic
		merciful
		selfless/altruistic
		generous
		supportive
		understanding
	uncaring/callous	self-absorbed
		selfish/self-serving
		merciless/ruthless
EXPECTATION	optimistic	positive
		sanguine
	pessimistic	negative/defeatist
		cynical
EXTROVERSION/ INTROVERSION	outgoing/extroverted	uninhibited/unreserved
		sociable
		approachable
DEPENDENCE	dependent	helpless
		obedient
		servile/obsequious
	independent	confident

		responsible/trustworthy / dependable
MOTIVATION	motivated	inspired
		excited/stimulated
	apathetic/indifferent	dismissive
		discouraged/dejected
OPENNESS/TRUST	open	honest/sincere
		candid/frank
		reasonable
		trusting
	trustworthy/responsible/ dependable	faithful/loyal
	closed/distant	distrustful
		dishonest/deceitful
PRAISING/CRITICISM	laudatory	congratulatory
		flattering
	critical	belittling/contemptuous
RESENTMENT/ FORGIVENESS	forgiving	understanding
		merciful
	unforgiving/vindictive/spiteful/vengeful	petty
RESPONSIVENESS	responsive/demonstrative	enthusiastic
		emotional/passionate
	Unresponsive/undemonstrative	unenthusiastic
		unemotional/detached
		dispassionate
SELF-PROMOTION	boastful	pompous/pretentious
	modest/humble/unassuming	self-deprecating/self-effacing
SELF-ESTEEM	conceited/vain	smug
	self-deprecating/self-effacing	
SEXUALITY	seductive	suggestive/risqué/ naughty
	lewd/bawdy/indecent	
	prudish/priggish	
SOCIAL DOMINANCE/ CONFIDENCE	arrogant	forward/presumptuous
		brazen
		commanding/ domineering
		condescending/ patronizing/ snobbish

		pedantic
		pompous/pretentious
	confident	cool
	submissive	servile/obsequious
	obedient	
	rebellious/defiant	
SOCIAL RANK	polite/courteous/respectful	unaffected
	rude/disrespectful	

Table 56 provides the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns above.

Table 2 – Basic Social Attitude Semantics Set

ID	Social Attitude	Meaning
1	accepting	attitude communicating willingness to accept into relationship or group
2	admiring/approving	attitude due to perception that others' actions or results are valuable
3	aggressive	tending to physically or metaphorically attack
4	apathetic/indifferent	showing lack of interest
5	approachable	sociable and not inspiring inhibition
6	argumentative	tending to argue or dispute
7	arrogant	emotion communicating social dominance
8	assertive	taking active role in social situations
9	awed	approval combined with incomprehension or fear
10	belittling	criticising by understating victim's achievements, personal attributes, etc.
11	boastful	tending to praise or promote self
12	brazen	high degree of forwardness/presumption
13	candid/frank	open in linguistic communication
14	closed/distant	not open
15	commanding/domineering	tending to assert right to command
16	combative/belligerent	high degree of aggression, often physical
17	communicative	evincing willingness to communicate as needed
18	conceited/vain	evincing undesirable degree of self-esteem
19	condescending / patronizing / snobbish	disrespectfully asserting superior social status, experience, knowledge, or membership
20	confident	attitude due to belief in own ability
21	congratulatory	wishing well related to another's success or good luck
22	contemptuous	high degree of disapproval and perceived superiority
23	controlling	undesirably assertive

24	cool	repressing outward reaction, often to indicate confidence or dominance, especially when confronting aggression, panic, etc.
25	cooperative/agreeable	communicating willingness to cooperate
26	critical	attitude expressing disapproval
27	cynical	habitually negative, reflecting disappointment or disillusionment
28	dependent	evincing inability to function without aid
29	discouraged/dejected	unmotivated because goals or rewards were not achieved
30	disagreeable	not agreeable
31	disapproving	not approving
32	dishonest/deceitful/insincere	not honest
33	dismissive	actively indicating lack of interest or motivation
34	distrustful	not trusting
35	emotional/passionate	high degree of responsiveness to emotions
36	empathetic/caring	interested in or vicariously feeling others' emotions
37	enthusiastic	high degree of positive response, especially to specific occurrence
38	excited/stimulated	attitude indicating cognitive and emotional arousal
39	exclusive/cliquish	not welcoming into a social group
40	flattering	praising with intent to influence, often insincere
41	flexible	willing to adjust to changing circumstances or needs
42	forward/presumptuous	not observing norms related to intimacy or rank
43	forgiving	tending to forgive improper behaviour
44	friendly	welcoming or inviting social contact
45	generous	tending to give to others, materially or otherwise
46	guilty/remorseful/sorry	regret due to consciousness of hurting or damaging others
47	helpless	high degree of dependence
48	honest/sincere	tending to communicate without deception
49	independent	not dependent
50	indifferent	neither approving nor disapproving
51	inhibited/ reserved/ introverted/ withdrawn	unable or unwilling to participate socially
52	inspired	motivated by some person, event, etc.
53	irresponsible	not responsible
54	kind	tending to act as motivated by empathy or sympathy
55	laudatory	praising
56	lewd/bawdy/indecent	evoking sexual associations in ways beyond social norms
57	like-minded	attitude expressing agreement

58	melodramatic	high or excessive degree of responsiveness or demonstrativeness
59	merciful	tending to avoid punishing others, often motivated by empathy or sympathy
60	merciless/ruthless	not merciful
61	mocking	communicating non-physical aggression, often by imitating a disapproved aspect of the victim
62	modest/humble/unassuming	not boastful
63	motivated	communicating goal-directed emotion and cognitive state
64	negative/defeatist	expressing pessimism, often habitually
65	obedient	evincing tendency to obey commands
66	open	tending to communicate without inhibition
67	optimistic	tending to expect positive events or results
68	outgoing/ extroverted/ uninhibited/ unreserved	not inhibited
69	passive	not assertive
70	passive-aggressive	covertly and non-physically aggressive
71	peaceful	not aggressive
72	pedantic	excessively displaying knowledge or academic status
73	permissive	allowing activity that social norms might restrict
74	pessimistic	tending to expect negative events or results
75	petty	unforgiving concerning small matters
76	polite/courteous/respectful	tending to respect social norms
77	pompous/pretentious	excessively displaying social rank, often above actual status
78	positive	expressing optimism, often habitually
79	prudish/priggish	expressing disapproval of even minor social transgressions, especially related to sex
80	reasonable	evincing willingness to resolve issues through reasoning
81	rebellious/defiant	evincing unwillingness to obey
82	responsible/trustworthy/ dependable	evincing characteristics or behaviour that encourage trust
83	responsive/demonstrative	tending to outwardly react to emotions and cognitive states, often as prompted by others
84	rude/disrespectful	not polite or respectful
85	sanguine	low degree of optimism, often expressed calmly
86	sarcastic	communicating disagreement by pretending agreement in an obviously insincere manner
87	seductive	communicating interest in sexual or related contact
88	self-absorbed	not empathetic due to excessive interest in self
89	self-deprecating/self-effacing	tending to criticize, or fail to praise or promote, self

90	selfish/self-serving	not generous due to excessive interest in own benefit
91	selfless/altruistic	tending to act for others' benefit, sometimes exclusively
92	servile/obsequious	excessively and demonstrably obedient
93	shy	low degree of social inhibition
94	smug	evincing undesirable degree of self-esteem related to perceived triumph
95	stubborn	unwilling to change one's mind or behaviour
96	sociable	comfortable in social situations
97	submissive	tending to submit to social dominance
98	subversive/undermining	communicating intention to work against a victim's goals
99	suggestive/risqué/naughty	evoking sexual associations within social norms
100	supportive	communicating willingness to support as needed
101	sympathetic	empathetic related to others' hurt or suffering
102	trusting	tending to trust others
103	unaffected	not pompous
104	uncaring/callous	not empathetic or caring
105	uncommunicative	not communicative
106	uncooperative	not cooperative
107	understanding	forgiving due to ability to understand motivations
108	unemotional/dispassionate/detached	not emotional, even when emotion is expected
109	unenthusiastic	not enthusiastic
110	unfriendly/hostile	not friendly
111	unresponsive/undemonstrative	not responsive or demonstrative
112	welcoming/inviting	high degree of acceptance with emotional warmth

The appropriate MPAI Development Committee will examine the proposed extension or replacement. Only the adequacy of the proposed new tables in terms of clarity and completeness will be considered. In case the new tables are not clear or complete, a revision of the tables will be requested.
The accepted Social Attitude Set will be identified as proposed by the submitter and reviewed by the appropriate MPAI Committee and posted to the MPAI web site.
The versioning system is based on a name – MPAI for MPAI-generated versions or "organisation name" for the proposing organisation – with a suffix m.n where m indicates the version and n indicated the subversion.

8.7.3 Syntax

https://schemas.mpai.community/MMC/V2.4/data/SocialAttitude.json

8.7.4 Semantics

Label	Description
Header	Entity Social Attitude Header
- Standard-SocialAttitude	The characters “MMC-ESA-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SocialAttitudeID	Identifier of the Social Attitude.
SocialAttitudeSpaceTime	Space-Time info of Social Attitude.
SocialAttitudeData	Data associated to Social Attitude.
- FusedSocAtt	Integrated Social Attitude Value.
- TextSocAtt	Text Social Attitude Value.
- SpeechSocAtt	Speech Social Attitude Value.
- FaceSocAtt	Face Social Attitude Value.
- GestureSocAtt	Gesture Social Attitude Value.
DescrMetadata	Descriptive Metadata

8.7.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Entity Social Attitude (MMC-ESA) if:

1. The Data validates against the Entity Social Attitude’s JSON Schema.
2. All Data in the Entity Social Attitude’s JSON Schema
    1. Have the specified type.
    2. Validate against their JSON Schemas.
    3. Conform with their Data Qualifiers if present.

8.8 Speech Descriptors

8.8.1 Definition

A Data Type representing characteristic elements extracted from the input speech, specifically Pitch, Intensity, Tempo, Personal Status, and NNSpeechFeatures in a period of time.

8.8.2 Functional Requirements

Speech Descriptors may include Neural Network Descriptors.

8.8.3 Syntax

https://schemas.mpai.community/MMC/V2.4/data/SpeechDescriptors.json

8.8.4 Semantics

Label	Description
Header	Speech Descriptors Header
- Standard - SpeechDescriptors	The characters "MMC-SPD-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	ID of the Metaverse Instance.
SpeechDescriptorsID	ID of Speech Descriptors.
SpeechDescriptorsData	Data associated with Input Text.
NNSpeechFeatures	The output vector of a neural-network using Speech as input.
Duration	The Time in which the Speech Descriptors are computed.
Pitch	Real number measuring the fundamental frequency of Speech in Hz (Hertz).
Intensity	Real number measuring the Energy of Speech in dBs (decibel).
Tempo	Real number measuring the rate at which specified linguistic units (Phonemes, Syllables, or Words) are produced.
Personal Status	The Speech Personal Status carried by the input speech.

8.8.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Speech Descriptors (MMC-SPD) if:

1. The Data validates against the Speech Descriptors' JSON Schema.
2. All Data in the Speech Descriptors' JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers if present.

8.9 Text Descriptors

8.9.1 Definition

A Data Type representing the syntactic and semantic information of a Text.

8.9.2 Functional Requirements

Meaning is an extract of the information from text to help an Entity Dialogue Processing AIM to produce a response.

8.9.3 Syntax

https://schemas.mpai.community/MMC/V2.4/data/TextDescriptors.json

8.9.4 Semantics

Label	Description
Header	Text Descriptors Header
- Standard - TextDescriptors	The characters “MMC-TXD-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
TextDescriptorsID	ID of Text Descriptors
TextDescriptors	Identifier of the AV Object.
- POS_tagging	Results of POS (Part of Speech, e.g., noun, verb, etc.) tagging including information on the question’s POS tagging set and tagged results.
- NE_tagging	Results of NE (Named Entity e.g., Person, Organisation, Fruit, etc.) tagging results including information on the question’s tagging set and tagged results.
- Dependency_tagging	Results of dependency (structure of the sentence, e.g., subject, object, head of relation, etc.) tagging including information on the question’s dependency tagging set and tagged results.
- SRL_tagging	Results of SRL (Semantic Role Labelling) tagging results including information on the question’s SRL tagging set and tagged results. SRL indicates the semantic structure of the sentence such as agent, location, patient role, etc.
DesrMetadata	Descriptive Metadata

8.9.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.4 Text Descriptors (MMC-TXD) if:

1. The Data validates against the Text Descriptors’ JSON Schema.
2. All Data in the Text Descriptors’ JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers if present.

8.10 3D Model Object

8.10.1 Definition

A Data Type including a collection of Basic 3D Model Objects.
A 3D Model Object can have a hierarchical structure where 3D Model Objects contain Basic 3D Model Objects and 3D Model Objects.

8.10.2 Functional Requirements

A 3D Model Object may include:

1. ID of a Virtual Space (M-Instance) where it is or intended to be located.

2. ID of the 3D Model Object.
3. Space-Time information of the 3D Model Object.
4. Basic 3D Model Objects and other 3D Model Objects included in the 3D Model Objects.
5. Annotation data set including:
    1. Annotations
    2. Space-Times of the Annotations.
    3. Rights to perform Actions on the 3D Model Object Annotation.
6. The Rights that may be exercised on the 3D Model Object.

Note that.

1. A 3D Model Object that does not include 3D Model Objects and only one Basic 3D Model Object is a Basic 3D Model Object.
2. The Space-Time information of a Basic 3D Model Object and a 3D Model Object included in an 3D Model Object may be superseded by the Space-Time information of the 3D Model Object containing them.

8.10.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/3DModelObject.json

8.10.4 Semantics

Label	Description
Header	3D Model Object Header
– Standard-3DModelObject	The characters “OSD-3DO-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
3DModelObjectID	Identifier of the 3D Model Object.
3DModelObjectSpaceTime	Space-Time of 3D Model Object.
Basic3DModelObjectCount	Set of Parent 3D Model Objects.
Basic3DModelObjects[]	Set of Basic 3D Model Objects.
- SpaceTime	Space Time of a Basic 3D Model Object in the 3D Model Object.
- Basic3DModelObject	A Basic 3D Model Object in the 3D Model Object.
3DModelObjectCount	Number of 3D Model Objects.
3DModelObjects[]	Set of 3D Model Objects.
- SpaceTime	Space Time of an 3D Model Object in the 3D Model Object.
- 3DModelObject	A 3D Model Object in the 3D Model Object
Annotations[]	Set of 3D Model Object Annotation.
– Annotation	An Annotation.
– AnnotationSpaceTime	Where Annotation is attached and when it will be active.
– Rights	Process Actions that may be performed on the Annotation
Rights	Process Actions that may be performed on the Object.
DescrMetadata	Descriptive Metadata

8.10.5 Conformance Testing

A Data instance Conforms with 3D Model Object (OSD-3DO) if:

1. The Data validates against the 3D Model Object's JSON Schema.
2. All Data in the 3D Model Object's JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers.

8.11 Audio Object

8.11.1 Definition

A Data Type including Basic Audio Objects and Audio Objects.
Audio Objects have a hierarchical structure where Audio Objects contain Basic Audio Objects and Audio Objects.

8.11.2 Functional Requirements

An Audio Object may include:

1. ID of a Virtual Space (M-Instance) where it is or intended to be located.
2. ID of the Audio Object.
3. Space-Time information of the Audio Object.
4. Basic Audio Object and Audio Objects included in the Audio Objects.
5. Annotation data set including:
    1. Annotations
    2. Space-Times of the Annotations.
    3. Rights to perform Actions on the Audio Object.
6. The Rights that may be exercised on the Audio Object.

Note that.

1. An Audio Object that does not include Sub-Scenes and only one Basic Audio Object is a Basic Audio Object.
2. The Space-Time information of a Basic Audio Object, Audio Object included in an Audio Object may be superseded by the Space-Time information of the Audio Object containing it.

Syntax

https://schemas.mpai.community/OSD/V1.4/data/AudioObject.json

8.11.3 Semantics

Label	Description
Header	Audio Object Header
– Standard-AudioObject	The characters "OSD-AUO-V"
– Version	Major version – 1 or 2 characters
– Dot-separator	The character "."
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
AudioObjectID	Identifier of the Audio Object.
AudioObjectSpaceTime	Space-Time of Audio Object.

BasicAudioObjectCount	Set of Parent Audio Objects.
BasicAudioObjects[]	Set of Basic Audio Objects.
- SpaceTime	Space Time of a Basic Audio Object in the Audio Object.
- BasicAudioObject	A Basic Audio Object in the Audio Object.
AudioObjectCount	Number of Audio Objects.
AudioObjects[]	Set of Audio Objects.
- SpaceTime	Space Time of an Audio Object in the Audio Object.
- AudioObject	An Audio Object in the Audio Object
Annotations[]	Set of Audio Object Annotation.
– Annotation	An Annotation.
– AnnotationSpaceTime	Where Annotation is attached and when it will be active.
– Rights	Actions that may be performed on the Annotation
Rights	Actions that may be performed on the Object.
DescrMetadata	Descriptive Metadata

8.11.4 Conformance Testing

A Data instance Conforms with Audio Object (OSD-AUO) if:

1. The Data validates against the Audio Object's JSON Schema.
2. All Data in the Audio Object's JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers.

8.12 Audio Scene Descriptors

8.12.1 Definition

Audio Scene Descriptors are defined as a Data Type including the Audio Objects of a scene, their sub-scenes, and their arrangement in the scene. Audio Scene Descriptors may be hierarchical, i.e., they may contain Objects and Audio Scene Descriptors.

8.12.2 Functional Requirements

Audio Scene Descriptors include

1. Audio Objects
2. The Descriptors of the Scenes includes in the Scene called Sub-Scenes.
3. Rights that may be exercised on the Scene.

Scenes may be hierarchical, i.e., they may contain Objects and Scenes.

8.12.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/AudioSceneDescriptors.json

8.12.4 Semantics

Label	Description
Header	Audio Scene Descriptors Header
- Standard-AudioSceneDescriptors	The characters “OSD-ASD-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SceneDescriptorsID	Identifier of Scene Descriptors.
SceneDescriptorsSpaceTime	Space and Time of Scene Descriptors.
ObjectCount	Number of Objects in Scene.
Objects[]	Set of Objects.
- Object or ObjectID	Object in the Scene of its ID.
- ObjectSpaceTime	Space Time of Object.
SubSceneCount	Number of Sub-Scenes in Scene.
SubScenes[]	Set of Sub-Scenes in the Scene.
- SubScene or SubSceneID	Sub-Scene in the Scene or its ID.
- SubSceneSpaceTime	Space Time of Sub-Scene.
DescrMetadata	Descriptive Metadata

8.12.5 Conformance Testing

A Data instance Conforms with Audio Scene Descriptors (OSD-ASD) if:

1. The Data validates against the Scene Descriptors’ JSON Schema.
2. All Data in the Scene Descriptors’ JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers.

8.13 Audio Scene Geometry

8.13.1 Definition

A Data Type including the space-time arrangement of the Audio Objects in a scene.
In the following, Data, Objects, Qualifiers, and (Sub-)Scenes should be read as Audio Data, Audio Objects, Audio Qualifiers, and Audio (Sub-)Scenes

8.13.2 Functional Requirements

Scene Geometry includes the arrangements of the Scenes - called Sub-Scenes - in addition to the arrangement of Objects.

8.13.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/AudioSceneGeometry.json

8.13.4 Semantics

Label	Description
Header	Audio Scene Geometry Header
- Standard-AudioSceneGeometry	The characters "OSD-ASG-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SceneGeometryID	Identifier of Scene Geometry.
ObjectCount	Number of Objects in Scene.
SubSceneCount	Number of Sub-Scenes in Scene.
SceneGeometrySpaceTime	Space and Time of Scene Geometry.
SceneObjects[]	Set of Data related to Objects.
- SceneObjectID	ID of Object.
- SceneObjectSpaceTime	Space Time of Object.
SceneSubScenes[]	Set of Sub-Scenes.
- SceneSubSceneID	ID of Sub-Scene.
- SceneSubSceneSpaceTime	Space Time of Sub-Scene.
DescrMetadata	Descriptive Metadata.

8.13.5 Conformance Testing

A Data instance Conforms with Audio Scene Geometry (OSD-ASG) if:

1. The Data validates against the Scene Geometry's JSON Schema.
2. All Data in the Scene Geometry's JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers.

8.14 Basic Audio-Visual Scene Descriptors

8.14.1 Definition

A Data Type including the Audio-Visual Objects of a scene, their time and arrangement in the scene, and the Rights that may be exercised on the scene.
In the following Object and Scene are to be read as Audio-Visual Object and Audio-Visual Scene, respectively.

8.14.2 Functional Requirements

Basic Scene Descriptors include

1. Objects
2. Space-Time information.
3. Rights that may be exercised on the Scene.

The Space-Time of the Objects may be superseded by the Space-Time of the Scene.

8.14.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/BasicAudioVisualSceneDescriptors.json

8.14.4 Semantics

Label	Description
Header	Basic Audio-Visual Scene Descriptors Header
- Standard-BasicAudioVisualSceneDescriptors	The characters "OSD-BMS-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SceneDescriptorsID	Identifier of Scene Descriptors.
ObjectCount	Number of Objects in Scene.
SceneDescriptorsSpaceTime	Space and Time of Scene Descriptors.
SceneObjects[]	Set of Objects.
- SceneObject	An Object.
- SceneObjectSpaceTime	Space Time of Object.
Rights	Rights that may be exercised on the Scene.
DescrMetadata	Descriptive Metadata

8.14.5 Conformance Testing

A Data instance Conforms with Basic Scene Descriptors (OSD-BMS) if:

1. The Data validates against the Scene Descriptors' JSON Schema.
2. All Data in the Scene Descriptors' JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers.

8.15 Basic Audio-Visual Scene Geometry

8.15.1 Definition

A Data Type including the arrangement of the 3D Model Objects in a scene.
In the following, Data, Objects, Qualifiers, and Scenes should be read as Audio-Visual Data, Audio-Visual Objects, Audio-Visual Qualifiers, and Audio-Visual Scenes.

8.15.2 Functional Requirements

Basic Scene Geometry includes the Qualifiers and the Space-Time of the Objects.

8.15.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/BasicAudioVisualSceneGeometry.json

8.15.4 Semantics

Label	Description
Header	Basic Audio-Visual Scene Geometry Header
- Standard-BasicAudioVisualSceneGeometry	The characters "OSD-BMG-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SceneGeometryID	Identifier of Scene Geometry.
ObjectCount	Number of Objects in Scene.
SceneGeometrySpaceTime	Space and Time of Scene Geometry.
SceneObjects[]	Set of Data related to Objects.
- SceneObjectQualifiers	Qualifiers of Object.
- SceneObjectSpaceTime	Space Time of Object.
DescrMetadata	Descriptive Metadata

8.15.5 Conformance Testing

A Data instance Conforms with Basic Audio-Visual Scene Geometry (OSD-BMG) if:

1. The Data validates against the Scene Geometry's JSON Schema.
2. All Data in the Scene Geometry's JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers.

8.16 Audio-Visual Event Descriptors

8.16.1 Definition

An Item including a series of Audio-Visual Scene Descriptors for a certain duration.

8.16.2 Functional Requirements

Audio-Visual Event Descriptors contains Audio-Visual Scene Descriptors for a Time.

8.16.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/AudioVisualEventDescriptors.json

8.16.4 Semantics

Label	Description
Header	Audio-Visual Event Descriptors Header
- Standard-AudioVisualEventDescriptors	The characters "OSD-AVE-V"
· Version	Major version – 1 or 2 characters
· Dot-separator	The character "."
· Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
EventID	Identifier of the Event.
EventSpaceTime	Data about start and end Space-Time.
SceneDescriptors[]	Set of Scene Descriptors
- SceneDescriptors	Set of AV Scene Descriptors of IDs.
DescrMetadata	Descriptive Metadata

8.16.5 Conformance Testing

A Data instance Conforms with MPAI-OSD Audio-Visual Event Descriptors (OSD-AVE) if:

1. The Data validates against the Annotation's JSON Schema.
2. All Data in the Annotation's JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers if present.

8.17 Audio-Visual Object

8.17.1 Definition

Data whose rendering has both Audio and Visual perceptibility attributes.

8.17.2 Functional Requirements

Audio-Visual Object includes:

1. The ID of a Virtual Space (M-Instance) where it is or will be located.
2. The 3DModel-Speech-Audio-Visual Objects' Space-Time location.
3. The IDs of the 3DModel, Speech, Audio, and Visual Objects' and their Space-Time information.

8.17.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/AudioVisualObject.json

8.17.4 Semantics

Label	Description
Header	Audio-Visual Object Header

- Standard-AudioVisualObject	The characters "OSD-AVO-V"
- Version	Major version – 1 or 2 Bytes
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 Bytes
MInstanceID	Identifier of M-Instance.
AudioVisualObjectID	Identifier of Audio-Visual Object.
AudioVisualObjectSpaceTime	Space-Time of Audio-Visual Object
AudioVisualQualifier	Qualifier of the Audio-Visual Object
3DModelObjectData	3D Model Object Data
- 3DModelObjectID and/or 3DModelObject	3D Model Object ID and/or Object
- 3DModelObjectSpaceTime	Space-Time of Speech Object
SpeechObjectData	Speech Object Data
- SpeechObjectID and/or Speech Object	Speech Object ID and/or Object
- SpeechObjectSpaceTime	Space-Time of Speech Object
AudioObjectData	Audio Object Data
- AudioObjectID and/or Audio Object	Audio Object ID and/or Object
- AudioObjectSpaceTime	Space-Time of Audio Object
VisualObjectData	Visual Object Data
- VisualObjectID and/or Visual Object	Visual Object ID and/or Object
- VisualObjectSpaceTime	Space-Time of Visual Object
Annotations[]	Set of Audio Object Annotation.
– Annotation	An Annotation.
– AnnotationSpaceTime	Where Annotation is attached and when it will be active.
– Rights	Actions that may be performed on the Annotation
Rights	Actions that may be performed on the Object.
DescrMetadata	Descriptive Metadata

8.17.5 Conformance Testing

A Data instance Conforms with Audio-Visual Object (OSD-AVO) if:

1. The Data validates against the Audio-Visual Object's JSON Schema.
2. All Data in the Audio-Visual Object's JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas

3. Conform with their Data Qualifiers if present.

8.18 Audio-Visual Scene Descriptors

8.18.1 Definition

A Data Type including the Audio-Visual Scene's Objects and Sub-Scenes and their arrangement in the Scene.

8.18.2 Functional Requirements

Audio-Visual Scene Descriptors includes Scenes in addition to Objects.

8.18.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/AudioVisualSceneDescriptors.json

8.18.4 Semantics

Label	Description
Header	Audio-Visual Scene Descriptors Header
- Standard-AVSceneDescriptors	The characters "OSD-AVS-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
AVBasicSceneDescriptorsID	Identifier of the AV Object.
ObjectCount	Number of Objects in Scene
AVSceneSpaceTime	Data about Space and Time
SpeechObjects[]	Set of Speech Objects
- SpeechObject	Speech Object
- SpeechObjectSpaceTime	Space-Time of Speech Object
AudioObjects[]	Set of Audio Objects
- AudioObject	ID of Audio Object
- AudioObjectSpaceTime	Space-Time of Audio Object
VisualObjects[]	Set of Visual Objects
- VisualObjectID	ID of Visual Object
- VisualObjectSpaceTime	Space-Time of Visual Object
AudioVisualObjects[]	Set of Audio-Visual Objects
- AudioVisualObjectID	ID of Audio-Visual Object
- AudioObjectSpaceTime	Space-Time of Audio-Visual Object
SubSceneCount	Number of Sub-Scenes in Scene
SpeechSubScenes[]	Set of Speech Objects
- SpeechSubScene	Speech SubScene
- SpeechSubSceneSpaceTime	Space-Time of Speech SubScene
AudioSubScenes[]	Set of Audio SubScenes

- AudioSubScene	ID of Audio SubScene
- AudioSubSceneSpaceTime	Space-Time of Audio SubScene
VisualSubScenes[]	Set of Visual SubScenes
- VisualSubSceneID	ID of Visual SubScene
- VisualSubSceneSpaceTime	Space-Time of Visual SubScene
AudioVisualSubScenes[]	Set of Audio-Visual SubScenes
- AudioVisualSubSceneID	ID of Audio-Visual SubScene
- AudioSubSceneSpaceTime	Space-Time of Audio-Visual SubScene
DescrMetadata	Descriptive Metadata

8.18.5 Conformance Testing

A Data instance Conforms with Audio-Visual Scene Descriptors (OSD-AVS) V1.3 if:

1. The Data validates against the Audio-Visual Scene Descriptors' JSON Schema.
2. All Data in the Audio-Visual Scene Descriptors' JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers if present.

8.19 Audio-Visual Scene Geometry

8.19.1 Definition

An Data Type including the arrangement of the Audio-Visual Objects in a scene with their Audio-Visual Qualifiers.

In the following, Data, Objects, Qualifiers, and (Sub-)Scenes should be read as Audio-Visual Data, Audio-Visual Objects, Audio-Visual Qualifiers, and Audio-Visual (Sub-)Scenes

8.19.2 Functional Requirements

Scene Geometry includes the arrangements of the Scenes - called Sub-Scenes - in addition to the arrangement of Objects.

8.19.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/AudioVisualSceneGeometry.json

8.19.4 Semantics

Label	Description
Header	Audio-Visual Scene Geometry Header
- Standard-AudioVisualSceneGeometry	The characters "OSD-AVG-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SceneGeometryID	Identifier of Scene Geometry.

ObjectCount	Number of Objects in Scene.
SubSceneCount	Number of Sub-Scenes in Scene.
SceneGeometrySpaceTime	Space and Time of Scene Geometry.
SceneObjects[]	Set of Data related to Objects.
– SceneObjectID	ID of Object.
– SceneObjectSpaceTime	Space Time of Object.
SceneSubScenes[]	Set of Sub-Scenes.
– SceneSubSceneID	ID of Sub-Scene.
– SceneSubSceneSpaceTime	Space Time of Sub-Scene.
DescrMetadata	Descriptive Metadata.

8.19.5 Conformance Testing

A Data instance Conforms with Audio-Visual Scene Geometry (OSD-AVG) if:

1. The Data validates against the Scene Geometry's JSON Schema.
2. All Data in the Scene Geometry's JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers.

8.20 Instance Identifier

8.20.1 Definition

A Data Type associating a string (Identifier) with an element of a set of entities – Speech, Objects, Visual Objects, User IDs etc. – belonging to some levels in a hierarchical classification (taxonomy).

8.20.2 Functional Requirements

Instance Identifier includes:

1. ID of Virtual Space (M-Instance)
2. Instance Label
3. Confidence level of the association between Instance Label and Instance.
4. Taxonomy
5. Confidence level of the association between Taxonomy and the Instance.

8.20.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/InstanceIdentifier.json

8.20.4 Semantics

Label	Description
Header	Instance Identifier Header
– Standard-InstanceIdentifier	The characters "OSD-IID-V"
– Version	Major version – 1 or 2 characters
– Dot-separator	The character "."
– Subversion	Minor version – 1 or 2 characters

MInstanceID	Identifier of M-Instance
InstanceID	Identifier of Instance.
InstanceSpaceTime	Data about Space-Time
InstanceIdentifierData	Data set of Instance Identifier.
InstanceLabel	Instance identified by Instance Identifier.
LabelConfidenceLevel	Confidence of Instance Label and Instance association.
TaxonomyLabel	Taxonomy Instance Identifier belongs to.
TaxonomyConfidenceLevel	Confidence of Taxonomy Label .
TaxonomyDataLength	Number of Bytes
TaxonomyDataURI	URI of Taxonomy.
DescrMetadata	Descriptive Metadata

8.20.5 Conformance Testing

A Data instance Conforms with Instance Identifier (OSD-IID) if:

1. The Data validates against the Instance Identifier's JSON Schema.
2. All Data in the Instance Identifier's JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers if present.

8.21 Point of View

8.21.1 Definition

Position and Orientation of an Object in a Virtual Environment excluding velocity and acceleration.

8.21.2 Functional Requirements

- An Object may have one of the following attributes: Speech, Audio; Visual; 3D Model, Audio-Visual; Haptic; Smell; RADAR; LiDAR; Ultrasound.
- Accuracy is the estimated absolute difference between the measured spatial and angular values of each of CartPosition, SpherPosition, Orientation, and their true value.

8.21.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/PointOfView.json

8.21.4 Semantics

Table 1 provides the semantics of the components of Point of View. The following should be noted:

1. Each of Position, Velocity, and Acceleration is provided either in Cartesian (X,Y,Z) or Spherical (r,φ,θ) Coordinates.
2. The Euler angles are indicated by (α,β,γ).

Table 1 – Semantics of Point of View

Label	Description
Header	Point of View Header

- Standard-Point of View	The characters “OSD-OPV-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstance	ID id Virtual space Orientation refers tu
PointOfViewID	Identifier of Object Point of View.
General	Set of general data.
- CoordType	One of Cartesian, Spherical, Geodesic, Toroidal.
- ObjectType	One of Digital Human, Generic.
- MediaType	One of Speech, Audio, Visual, Audio-Visual, Haptic, Smell, RADAR, LiDAR, Ultrasound.
PositionAndOrientation	
- CartPosition (X,Y,Z)	Array (in metres)
- CartPositionAccuracy (X,Y,Z)	Array Of CartPositionAccuracy
- SpherPosition (r,φ,θ)	Array (in metres and degrees)
- SpherPositionAccuracy (r,φ,θ)	Array of - SpherPositionAccuracy
- Orient (α,β,γ)	Array (in degrees)
- OrientAccuracy (α,β,γ)	Array of OrientAccuracy
DescrMetadata	Descriptive Metadata

8.21.5 Conformance Testing

A Data instance Conforms with MPAI-OSD Point of View (OSD-OPV) if:

1. The Data validates against the Point of View’s JSON Schema.
2. All Data in the Point of View’s JSON Schema.
    1. Have the specified type.
    2. Validate against their JSON Schemas.

8.22 Selector

8.22.1 Definition

A Data Type used to indicate specific operating values of an AIW or AIM.

8.22.2 Functional Requirements

Selector informs an AIW/AIM that a communicating Entity uses/requests to use:

1. Specific media – Text, Speech, Visual, or Gesture – as input or output.
2. Specific Language – as input or output.
3. Media or their Descriptors.
4. View an Avatar or a Scene

8.22.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/Selector.json

8.22.4 Semantics

Label	Description
Header	Selector Header
- Standard-Selector	The characters “OSD-SEL-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
InputMedia	One or more of Text, Speech, Visual, or Gesture.
OutputMedia	One or more of Text, Speech, Visual, or Gesture.
InputLanguage	One of a list of languages.
OutputLanguage	One of a list of languages.
MediaOrDescriptors	One of Text, Speech, Face, Body for MMC-TST
SpeechDescriptors	One of No, Yes for MMC-PSE
View	One of Avatar or Scene
DescrMetadata	Descriptive Metadata

8.22.5 Conformance Testing

A Data instance Conforms with Selector (OSD-SEL) if:
1. The Data validates against the Selector’s JSON Schema.
2. All Data in the Selector’s JSON Schema have the specified types.

8.23 Space-Time

8.23.1 Definition

Data Type representing the Spatial Attitude and Time information.

8.23.2 Functional Requirements

Space-Time includes Spatial Attitude and Time.

8.23.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/SpaceTime.json

8.23.4 Semantics

Label	Description
Header	Space-Time Header
- Standard-Object	The characters “OSD-SPT-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters

MInstance	Identifier of Virtual Space.
SpaceTimeID	Identifier of Space-Time.
Space	Spatial Attitudes at T_0 and T_1
Time	Time interval between T_0 and T_1
DescrMetadata	Descriptive Metadata

8.23.5 Conformance Testing

A Data instance Conforms with Space-Time (OSD-SPT) if:

1. The Data validates against the Space-Time's JSON Schema.
2. All Data in the Space-Time's JSON Schema
    1. Have the specified type.
    2. Validate against their JSON Schemas.
    3. Conform with their Data Qualifiers if present.

8.24 Spatial Attitude

8.24.1 Definition

An Item representing the Position and Orientation of an Object, and their velocities and accelerations.

8.24.2 Functional Requirements

The Spatial Attitude is defined as the combination of Position and orientation, the Functional Requirements are defined by Position and Orientation.

8.24.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/SpatialAttitude.json

8.24.4 Semantics

Table 1 provides the semantics of the components of the Spatial Attitude.

Table 1 – Semantics of the Spatial Attitude

Label	Description
Header	Spatial Attitude Header
- Standard-SpatialAttitude	The characters "OSD-OSA-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	ID of Virtual Space Object refers to.
ObjectSpatialAttitudeID	Identifier of Object Spatial Attitude.
General	Set of general data
- CoordinateType	One of Cartesian, Spherical, Geodesic, Toroidal.
- ObjectType	One of Digital Human, Generic.

- MediaType	One of Speech, Audio, Visual, Audio-Visual, Haptic, Smell, RADAR, LiDAR, Ultrasound.
Position	As specified by Position
Orientation	As specified by Orientation
DescrMetadata	Descriptive Metadata

8.24.5 Conformance Testing

A Data instance Conforms with V1.2 Spatial Attitude (OSD-OSA) if:

1. The Data validates against the Spatial Attitude's JSON Schema.
2. All Data in the Spatial Attitude 's JSON Schema have the specified type.

8.25 Speech Object

8.25.1 Definition

A Data Type including a collection of Basic Speech Objects.
A Speech Object can have a hierarchical structure where Speech Objects contain Basic Speech Objects and Speech Objects.

8.25.2 Functional Requirements

A Speech Object may include:

1. ID of a Virtual Space (M-Instance) where it is or intended to be located.
2. ID of the Speech Object.
3. Space-Time information of the Speech Object.
4. Basic Speech Object and Speech Objects included in the Speech Objects.
5. Annotation data set including:
    1. Annotations
    2. Space-Times of the Annotations.
    3. Rights to perform Actions on the Speech Object.
6. The Rights that may be exercised on the Speech Object.

Note that.

1. A Speech Object that does not include Sub-Scenes and only one Basic Speech Object is a Basic Speech Object.
2. The Space-Time information of a Basic Speech Object and Speech Object included in a Speech Object may be superseded by the Space-Time information of the Speech Object containing them.

8.25.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/SpeechObject.json

8.25.4 Semantics

Label	Description
Header	Speech Object Header
– Standard-SpeechObject	The characters "OSD-SPO-V"
– Version	Major version – 1 or 2 characters
– Dot-separator	The character "."
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.

SpeechObjectID	Identifier of the Speech Object.
SpeechObjectSpaceTime	Space-Time of Speech Object.
BasicSpeechObjectCount	Set of Parent Speech Objects.
BasicSpeechObjects[]	Set of Basic Speech Objects.
- SpaceTime	Space Time of a Basic Speech Object in the Speech Object.
- BasicSpeechObject	A Basic Speech Object in the Speech Object.
SpeechObjectCount	Number of Speech Objects.
SpeechObjects[]	Set of Speech Objects.
- SpaceTime	Space Time of a Speech Object in the Speech Object.
- SpeechObject	A Speech Object in the Speech Object
Annotations[]	Set of Speech Object Annotation.
– Annotation	An Annotation.
– AnnotationSpaceTime	Where Annotation is attached and when it will be active.
– Rights	Actions that may be performed on the Annotation
Rights	Actions that may be performed on the Object.
DescrMetadata	Descriptive Metadata

8.25.5 Conformance Testing

A Data instance Conforms with Speech Object (OSD-SPO) if:

1. The Data validates against the Speech Object's JSON Schema.
2. All Data in the Speech Object's JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers.

8.26 Speech Scene Descriptors

8.26.1 Definition

A Data Type including the Speech Objects of a scene, their sub-scenes, and their arrangement in the scene.

8.26.2 Functional Requirements

Speech Scene Descriptors include

1. Speech Objects
2. The Descriptors of the Speech Scenes includes in the Speech Scene called Speech Sub-Scenes.
3. Rights that may be exercised on the Speech Scene.

Scenes may be hierarchical, i.e., they may contain Speech Objects and Speech Scenes.

8.26.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/SpeechSceneDescriptors.json

8.26.4 Semantics

Label	Description
Header	Speech Scene Descriptors Header

- Standard-SpeechSceneDescriptors	The characters "OSD-SSD-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SceneDescriptorsID	Identifier of Scene Descriptors.
SceneDescriptorsSpaceTime	Space and Time of Scene Descriptors.
ObjectCount	Number of Objects in Scene.
Objects[]	Set of Objects.
- Object or ObjectID	Object in the Scene of its ID.
- ObjectSpaceTime	Space Time of Object.
SubSceneCount	Number of Sub-Scenes in Scene.
SubScenes[]	Set of Sub-Scenes in the Scene.
- SubScene or SubSceneID	Sub-Scene in the Scene or its ID.
- SubSceneSpaceTime	Space Time of Sub-Scene.
DescrMetadata	Descriptive Metadata

8.26.5 Conformance Testing

A Data instance Conforms with Speech Scene Descriptors (OSD-SSD) if:

1. The Data validates against the Scene Descriptors' JSON Schema.
2. All Data in the Scene Descriptors' JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers.

8.27 Speech Scene Geometry

8.27.1 Definition

A Data Type including the arrangement of the Speech Objects in a scene with their Speech Qualifiers.
In the following, Data, Objects, Qualifiers, and (Sub-)Scenes should be read as Speech Data, Speech Objects, Speech Qualifiers, and Speech (Sub-)Scenes

8.27.2 Functional Requirements

Scene Geometry includes the arrangements of the Scenes - called Sub-Scenes - in addition to the arrangement of Objects.

8.27.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/SpeechSceneGeometry.json

8.27.4 Semantics

Label	Description
Header	Speech Scene Geometry Header
- Standard-SpeechSceneGeometry	The characters "OSD-SSG-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SceneGeometryID	Identifier of Scene Geometry.
ObjectCount	Number of Objects in Scene.
SubSceneCount	Number of Sub-Scenes in Scene.
SceneGeometrySpaceTime	Space and Time of Scene Geometry.
SceneObjects[]	Set of Data related to Objects.
– SceneObjectID	ID of Object.
– SceneObjectSpaceTime	Space Time of Object.
SceneSubScenes[]	Set of Sub-Scenes.
– SceneSubSceneID	ID of Sub-Scene.
– SceneSubSceneSpaceTime	Space Time of Sub-Scene.
DescrMetadata	Descriptive Metadata.

8.27.5 Conformance Testing

A Data instance Conforms with Speech Scene Geometry (OSD-SSG) if:

1. The Data validates against the Scene Geometry's JSON Schema.
2. All Data in the Scene Geometry's JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers.

8.28 Text Object

8.28.1 Definition

A Data Type including a collection of Basic Text Objects.
A Text Object can have a hierarchical structure where Text Objects contain Basic Text Objects and Text Objects.

8.28.2 Functional Requirements

A Text Object may include:

1. ID of a Virtual Space (M-Instance) where it is or intended to be located.
2. ID of the Text Object.
3. Space-Time information of the Text Object.
4. Basic Text Object and Text Objects included in the Text Objects.

5. Annotation data set including:
    1. Annotations
    2. Space-Times of the Annotations.
    3. Rights to perform Actions on the Text Object.
6. The Rights that may be exercised on the Text Object.

Note that.

1. A Text Object that does not include Sub-Scenes and only one Basic Text Object is a Basic Text Object.
2. The Space-Time information of a Basic Text Object and Text Object included in a Text Object may be superseded by the Space-Time information of the Text Object containing them.

8.28.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/TextObject.json

8.28.4 Semantics

Label	Description
Header	Text Object Header
– Standard-TextObject	The characters “OSD-TXO-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
TextObjectID	Identifier of the Text Object.
TextObjectSpaceTime	Space-Time of Text Object.
BasicTextObjectCount	Set of Parent Text Objects.
BasicTextObjects[]	Set of Basic Text Objects.
- SpaceTime	Space Time of a Basic Text Object in the Text Object.
- BasicTextObject	A Basic Text Object in the Text Object.
TextObjectCount	Number of Text Objects.
TextObjects[]	Set of Text Objects.
- SpaceTime	Space Time of a Text Object in the Text Object.
- TextObject	A Text Object in the Text Object
Annotations[]	Set of Text Object Annotation.
– Annotation	An Annotation.
– AnnotationSpaceTime	Where Annotation is attached and when it will be active.
– Rights	Actions that may be performed on the Annotation
Rights	Actions that may be performed on the Object.
DescrMetadata	Descriptive Metadata

8.28.5 Conformance Testing

A Data instance Conforms with Text Object (OSD-TXO) if:

1. The Data validates against the Text Object’s JSON Schema.
2. All Data in the Text Object’s JSON Schema

1. Have the specified type
2. Validate against their JSON Schemas
3. Conform with their Data Qualifiers.

8.29 Time

8.29.1 Definition

The digital representation of duration of time.

8.29.2 Functional Requirements

Time includes the digital representation of time specified by MPAI or other Standard Setting Organisations as indicated by Time Qualifier

8.29.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/Time.json

8.29.4 Semantics

Label	Description
Header	Time Header
- Standard-Object	The characters "OSD-TIM-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance
TimeID	Identifier of M-Instance.
TimeData	Data about Time
TimeQualifier	End of Time.
DescrMetadata	Descriptive Metadata

8.29.5 Conformance Testing

A Data instance Conforms with MPAI-OSD Time (OSD-TIM) if:

1. The Data validates against the Times's JSON Schema.
2. All Data in JSON Times has the specified type.

8.30 Basic Visual Scene Descriptors

8.30.1 Definition

A Data Type including the Visual Objects of a scene, their time and arrangement in the scene, and the Rights that may be exercised on the scene.
In the following, "Object" and "Scene" are to be read as "Visual Object" and "Visual Scene", respectively.

8.30.2 Functional Requirements

Basic Visual Scene Descriptors include

1. Visual Objects
2. Space-Time information.
3. Rights that may be exercised on the Basic Visual Scene.

8.30.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/BasicVisualSceneDescriptors.json

8.30.4 Semantics

Label	Description
Header	Basic Visual Scene Descriptors Header
- Standard-BasicVisualSceneDescriptors	The characters "OSD-BVS-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
BasicVisualSceneDescriptorsID	Identifier of Basic Visual Scene Descriptors.
GravityValue	The value of Gravity in the Basic Scene measure in m/s^2.
VisualObjectCount	Number of Visual Objects in Basic Visual Scene.
SceneDescriptorsSpaceTime	Space and Time of Basic Visual Scene Descriptors.
BasicVisualSceneObjects[]	Set of Visual Objects.
- BasicVisualSceneObject	A Visual Object.
- BasicVisualSceneObjectSpaceTime	Space Time of Visual Object.
Gravity	Value of Gravity in Scene measured in m/s^2
Rights	Rights that may be exercised on the Basic Visual Scene.
DescrMetadata	Descriptive Metadata

8.30.5 Conformance Testing

A Data instance Conforms with Basic Visual Scene Descriptors (OSD-BVS) if:

1. The Data validates against the Basic Visual Scene Descriptors' JSON Schema.
2. All Data in the Basic Visual Scene Descriptors' JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Visual Data Qualifiers.

8.31 Basic Visual Scene Geometry

8.31.1 Definition

A Data Type including the arrangement of the Basic Visual Scene Geometry Objects in a scene. In the following, Data, Objects, Qualifiers, and Scenes should be read as Visual Data, Visual Objects, Visual Qualifiers, and Visual Scenes.

8.31.2 Functional Requirements

Basic Scene Geometry includes the Qualifiers and the Space-Time of the Objects.

8.31.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/BasicVisualSceneGeometry.json

8.31.4 Semantics

Label	Description
Header	Basic Visual Scene Geometry Header
- Standard-BasicVisualSceneGeometry	The characters "OSD-BVG-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SceneGeometryID	Identifier of Scene Geometry.
ObjectCount	Number of Objects in Scene.
SceneGeometrySpaceTime	Space and Time of Scene Geometry.
SceneObjects[]	Set of Data related to Objects.
- SceneObjectQualifiers	Qualifiers of Object.
- SceneObjectSpaceTime	Space Time of Object.
DescrMetadata	Descriptive Metadata

8.31.5 Conformance Testing

A Data instance Conforms with Basic Visual Scene Geometry (OSD-BVG) if:

1. The Data validates against the Scene Geometry's JSON Schema.
2. All Data in the Scene Geometry's JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas

8.32 Visual Object

8.32.1 Definition

A Data Type including a collection of Basic Visual Objects.

A Visual Object can have a hierarchical structure where Visual Objects contain Basic Visual Objects and Visual Objects.

8.32.2 Functional Requirements

A Visual Object may include:

1. ID of a Virtual Space (M-Instance) where it is or intended to be located.
2. ID of the Visual Object.
3. Space-Time information of the Visual Object.
4. Basic Visual Object and Visual Objects included in the Visual Objects.

5. Annotation data set including:
    1. Annotations
    2. Space-Times of the Annotations.
    3. Rights to perform Actions on the Visual Object.
6. The Rights that may be exercised on the Visual Object.

Note that.

1. A Visual Object that does not include Sub-Scenes and only one Basic Visual Object is a Basic Visual Object.
2. The Space-Time information of a Basic Visual Object and Visual Object included in a Visual Object may be superseded by the Space-Time information of the Visual Object containing them.

8.32.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/VisualObject.json

8.32.4 Semantics

Label	Description
Header	Visual Object Header
– Standard-VisualObject	The characters “OSD-VIO-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
VisualObjectID	Identifier of the Visual Object.
VisualObjectProperty	Properties of Visual Object - If any is present overrides the properties of component objects.
- MaterialProperty	Object has material consistence (0=No, 1=Yes)
- GravityProperty	Object is subject to gravity (0=No, 1-Yes)
VisualObjectSpaceTime	Space-Time of Visual Object.
BasicVisualObjectCount	Set of Parent Visual Objects.
BasicVisualObjects[]	Set of Basic Visual Objects.
- SpaceTime	Space Time of a Basic Visual Object in the Visual Object.
- BasicVisualObject	A Basic Visual Object in the Visual Object.
VisualObjectCount	Number of Visual Objects.
VisualObjects[]	Set of Visual Objects.
- SpaceTime	Space Time of a Visual Object in the Visual Object.
- VisualObject	A Visual Object in the Visual Object
Annotations[]	Set of Visual Object Annotation.
– Annotation	An Annotation.
– AnnotationSpaceTime	Where Annotation is attached and when it will be active.
– Rights	Actions that may be performed on the Annotation
Rights	Actions that may be performed on the Object.

DescrMetadata	Descriptive Metadata

8.32.5 Conformance Testing

A Data instance Conforms with Visual Object (OSD-VIO) if:

1. The Data validates against the Visual Object's JSON Schema.
2. All Data in the Visual Object's JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers.

8.33 Visual Scene Descriptors

8.33.1 Definition

A Data Type including the Visual Objects of a scene, their sub-scenes, and their arrangement in the scene.

8.33.2 Functional Requirements

Visual Scene Descriptors include

1. Visual Objects
2. The Descriptors of the Visual Scenes includes in the Visual Scene called Visual Sub-Scenes.
3. Rights that may be exercised on the Visual Scene.

Scenes may be hierarchical, i.e., they may contain Objects and Scenes.

8.33.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/VisualSceneDescriptors.json

8.33.4 Semantics

Label	Description
Header	Visual Scene Descriptors Header
- Standard-VisualSceneDescriptors	The characters "OSD-VSD-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
VisualSceneDescriptorsID	Identifier of Visual Scene Descriptors.
ObjectCount	Number of Visual Objects in Scene.
SubSceneCount	Number of Visual Scenes in Scene.
VisualSceneDescriptorsSpaceTime	Space and Time of Visual Scene Descriptors.
VisualSceneObjects[]	Set of Visual Objects.
- VisualSceneObject	Visual Object.
- VisualSceneObjectSpaceTime	Space Time of Visual Object.

VisualSceneSubScenes[]	Set of Visual Sub-Scenes.
- VisualSceneSubScene	Visual Sub-Scene.
- VisualSceneSubSceneSpaceTime	Space Time of Visual Sub-Scene.
DescrMetadata	Descriptive Metadata

8.33.5 Conformance Testing

A Data instance Conforms with Visual Scene Descriptors (OSD-VSD) if:

1. The Data validates against the Visual Scene Descriptors' JSON Schema.
2. All Data in the Visual Scene Descriptors' JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers.

8.34 Visual Scene Geometry

8.34.1 Definition

A Data Type including the arrangement of the Visual Objects in a scene with their Visual Qualifiers.
In the following, Data, Objects, Qualifiers, and (Sub-)Scenes should be read as Visual Data, Visual Objects, Visual Qualifiers, and Visual (Sub-)Scenes

8.34.2 Functional Requirements

Scene Geometry includes the arrangements of the Scenes - called Sub-Scenes - in addition to the arrangement of Objects.

8.34.3 Syntax

https://schemas.mpai.community/OSD/V1.4/data/VisualSceneGeometry.json

8.34.4 Semantics

Label	Description
Header	Visual Scene Geometry Header
- Standard-VisualSceneGeometry	The characters "OSD-VSG-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SceneGeometryID	Identifier of Scene Geometry.
ObjectCount	Number of Objects in Scene.
SubSceneCount	Number of Sub-Scenes in Scene.
SceneGeometrySpaceTime	Space and Time of Scene Geometry.
SceneObjects[]	Set of Data related to Objects.

– SceneObjectID	ID of Object.
– SceneObjectSpaceTime	Space Time of Object.
SceneSubScenes[]	Set of Sub-Scenes.
– SceneSubSceneID	ID of Sub-Scene.
– SceneSubSceneSpaceTime	Space Time of Sub-Scene.
DescrMetadata	Descriptive Metadata.

8.34.5 Conformance Testing

A Data instance Conforms with Visual Scene Geometry (OSD-VSG) if:

1. The Data validates against the Scene Geometry's JSON Schema.
2. All Data in the Scene Geometry's JSON Schema
    1. Have the specified type
    2. Validate against their JSON Schemas
    3. Conform with their Data Qualifiers.

8.35 Avatar

8.35.1 Definition

A Data Type that includes data characterising the Avatar, the Virtual Space in which the Avatar is located, and the Space-Time information of the Avatar in the Virtual Space

8.35.2 Functional Requirements

Avatar conveys the following information:

1. The ID of the Virtual Space (M-Instance).
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
    4. Process ID morphing the Model
    5. ClothID

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.

8.35.3 Syntax

https://schemas.mpai.community/PAF/V1.5/data/Avatar.json

8.35.4 Semantics

Label	Description
Header	Avatar Header.
– Standard-Avatar	The characters "PAF-AVT-V"
– Version	Major version
– Dot-separator	The character "."

– Subversion	Minor version
MInstanceID	ID of Virtual Space the Avatar belongs to.
AvatarSpaceTime	The inherent Space-Time info of the Avatar.
AvatarID	Identifier of Avatar.
AvatarData	Set of Avatar-related Data.
- AvatarModel	Model of Avatar.
- BodyDescriptorsObject	Avatar Body Descriptors Object.
- FaceDescriptorsObject	Avatar Face Descriptors Object of Avatar.
- ProcessID	ID of process morphing Model.
- ClothID	ID of cloth work by Avatar.
DescrMetadata	Descriptive Metadata

8.35.5 Conformance Testing

A Data instance Conforms with Avatar (PAF-AVT) V1.5 if:

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

8.36 Body Descriptors Object

8.36.1 Definition

Body Descriptors Object refers to

1. Body Descriptors Data representing a human or a humanoid.
2. Body Descriptors Data Qualifier specified by MPAI-TFA providing information on Sub-Types, Formats and Attributes of Body Descriptors Data.

8.36.2 Functional Requirements

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

8.36.3 Syntax

https://schemas.mpai.community/PAF/V1.5/data/BodyDescriptorsObject.json

8.36.4 Semantics

Label	Description
Header	Body Descriptors Object Header.
– Standard-BodyDescriptorsObject	The characters "OSD-BDO-V"
– Version	Major version – 1 or 2 characters
– Dot-separator	The character "."
– Subversion	Minor version – 1 or 2 characters
M-InstanceID	ID of the Virtual Space where the Body is located.
BodyDescriptorsObjectID	Identifier of Body Descriptors.
BodyDescriptorsSpaceTime	Space-Time information of Body in the Virtual Space

BodyDescriptorsData	Data of Body Descriptors
BodyDescriptorsQualifier	Qualifier of Body Descriptors
DescrMetadata	Descriptive Metadata

8.36.5 Conformance Testing

A Data instance Conforms with Body Descriptors Object (PAF-BDO) if

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

8.37 Face Descriptors Object

8.37.1 Definition

Face Descriptors Object refers to

1. Face Descriptors Data representing the features of the Face of an Entity.
2. Face Descriptors Data Qualifier specified by MPAI-TFA providing information on Sub-Types, Formats and Attributes of Body Descriptors Data.

8.37.2 Functional Requirements

The features depend on the motion of the muscles of the Face of an Entity.

8.37.3 Syntax

https://schemas.mpai.community/PAF/V1.5/data/FaceDescriptorsObject.json

8.37.4 Semantics

Label	Description
Header	Face Descriptors Object Header.
– Standard-FaceDescriptorsObject	The characters "OSD-FDO-V"
– Version	Major version – 1 or 2 characters
– Dot-separator	The character "."
– Subversion	Minor version – 1 or 2 characters
M-InstanceID	ID of the Virtual Space where the Face is located.
FaceDescriptorsObjectID	Identifier of Face Descriptors.
FaceDescriptorsSpaceTime	Space-Time information of Face in the Virtual Space
FaceDescriptorsData	Data of Face Descriptors
FaceDescriptorsQualifier	Qualifier of Face Descriptors
DescrMetadata	Descriptive Metadata

8.37.5 Conformance Testing

A Data instance Conforms with Face Descriptors Object (PAF-FDO) if

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

8.37.6 Face Action Units (informative)

The Face Actions Units of the Facial Action Coding System (FACS) were originally developed by Carl-Herman Hjortsjö, adopted by Paul Ekman and Wallace V. Friesen (1978) and updated by Ekman, Friesen, and Joseph C. Hager (2002). Each Action Unit is represented by an Action Unit ID. The provide a set of Face Descriptors.

AU	Description	Facial muscle generating the Action
1	Inner Brow Raiser	Frontalis, pars medialis
2	Outer Brow Raiser	Frontalis, pars lateralis
4	Brow Lowerer	Corrugator supercilii, Depressor supercilii
5	Upper Lid Raiser	Levator palpebrae superioris
6	Cheek Raiser	Orbicularis oculi, pars orbitalis
7	Lid Tightener	Orbicularis oculi, pars palpebralis
9	Nose Wrinkler	Levator labii superioris alaquae nasi
10	Upper Lip Raiser	Levator labii superioris
11	Nasolabial Deepener	Zygomaticus minor
12	Lip Corner Puller	Zygomaticus major
13	Cheek Puffer	Levator anguli oris (a.k.a. Caninus)
14	Dimpler	Buccinator
15	Lip Corner Depressor	Depressor anguli oris (a.k.a. Triangularis)
16	Lower Lip Depressor	Depressor labii inferioris
17	Chin Raiser	Mentalis
18	Lip Puckerer	Incisivii labii superioris and Incisivii labii inferioris
20	Lip stretcher	Risorius with platysma
22	Lip Funneler	Orbicularis oris
23	Lip Tightener	Orbicularis oris
24	Lip Pressor	Orbicularis oris
25	Lips part	Depressor labii inferioris or relaxation of Mentalis, or Orbicularis oris
26	Jaw Drop	Masseter, relaxed Temporalis and internal Pterygoid
27	Mouth Stretch	Pterygoids, Digastric
28	Lip Suck	Orbicularis oris
41	Lid droop	Relaxation of Levator palpebrae superioris
42	Slit	Orbicularis oculi
43	Eyes Closed	Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis
44	Squint	Orbicularis oculi, pars palpebralis
45	Blink	Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis
46	Wink	Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis

61	Eyes turn left	Lateral rectus, medial rectus
62	Eyes turn right	Lateral rectus, medial rectus
63	Eyes up	Superior rectus, Inferior oblique
64	Eyes down	Inferior rectus, Superior oblique

The eye motion Action Units (Eyes turn left/right and Eyes up/down) have an additional parameter representing the degrees of the motion of a line perpendicular to the eye with respect to the 0° of the resting position when the eye looks horizontally and perpendicular to the body. The value of the angle is expressed with 1 Byte where 0= 0° degrees for the resting position (0°) and 255 for 90°.

8.37.7 Mapping of Face Action Units to Personal Status (Informative)

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

Personal Status	Cognitive State	Emotion	Prototypical (and variant AUs)
Happy		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
https://mpai.community/standards/mpai-paf/v1-5/ai-modules/

8.38 Portable Avatar

8.38.1 Definition

A Data Type that includes:

1. A set of avatar-related Data: M-Instance ID, Avatar ID, Avatar 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.

8.38.2 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.

8.38.3 Syntax

https://schemas.mpai.community/PAF/V1.5/data/PortableAvatar.json

8.38.4 Semantics

Label	Description
Header	The Header of the Portable Avatar Data.
– Standard-PortableAvatar	The characters “PAF-PAV-V”
– Version	Major version
– Dot-separator	The character "."
– Subversion	Minor version
MInstanceID	The ID of the M-Instance.
PortableAvatarID	Identifier of the Portable Avatar.
PortableAvatarData	Set of Data related to Avatar.
- AudioVisualSceneSpaceTime	Space and Time info of AV Scene in M-Instance.
- AudioVisualSceneDescriptors	AV Scene Descriptors.
- AvatarSpaceTime	Space-Time of Avatar instance in AV Scene.
- Avatar	Avatar's Model and Face and Body Descriptors.
- LanguageSelector	Avatar's Language Preference.
- TextObject	Text associated with Avatar.
- SpeechObject	Set of Data related to Speech Object.
- SpeechModel	Neural Network Model for Speech Synthesis.
- PersonalStatus	Personal Status of Avatar.
DescrMetadata	Descriptive Metadata

8.38.5 Conformance Testing

A Data instance Conforms with Portable Avatar (PAF-PAV) V1.5 if:
1. JSON Data validate against the Portable Avatar 's JSON Schema.
2. All Data in the Portable Avatar 's JSON Schema
    1. Have the specified type.

2. Validate against their JSON Schemas.
3. Conform with their Data Qualifiers if present.

9 Profiles

HMC-CEC Profiles are defined based on the availability of six groups of processing capabilities:

Processing	Processed Data
Receive	Communication Items from a Machine or Audio-Visual Scenes from a real space.
Extract	Personal Status from Modalities (Text, Speech, Face, or Gesture).
Understands	Personal Status using or not spatial information.
Translates	Text using Meaning of Entity's Text.
Generates	Response including Text and Personal Status.
Renders	Response.

Table 1 defines the Attributes and Sub-Attributes of the HMC-CEC Profiles. The Sub-Attributes are expressed with three characters where the first two represent the medium followed by O representing Object:

1. The Audio-Visual Scene represent Text Object (TXO), Speech Object (SPO), Audio Object (AUO), Visual Object (VIO), 3D Model Object, and Portable Avatar (PAF) Sub-Attributes, respectively.
2. The Personal Status, Understanding, Translation, and Display Response represent Text (TXO), Speech (SPO), Face (FCO), and Gesture (GSO), respectively.

The PAF 1st-levelSub-Attribute (S-Attribute) includes the following 2nd-level SS-Attributes:

1. AVA: The Avatar.
2. LNG: the Avatar Language Preferences.
3. TXO: Text Object associated with the Entity.
4. NNM: Speech Model used to synthesise Speech.
5. SPO: Speech Object associated with the Avatar.
6. EPS: The Personal Status of the Avatar.
7. SPC: The Space (SPaCe) where the Avatar is embedded.

The SPC Sub-Attribute of Understanding represents Spatial Information, i.e., the capability of an HMC-CEC implementation to use Spatial Information.
The LNG Sub-Attribute represent the ISO 639 Set 3 three-letter code.

Table 1 - Attribute and Sub-Attribute Codes of HMC-CEC.

Attributes	Codes	S-Attribute Codes					SS-Attribute Code						
AV Scene	AVS	TXO	SPO	AUO	VIO	PAF	AVA	LNG	TXO	NNM	SPO	EPS	SPG
Personal Status	EPS	TXO	SPO	FCO	GSO								
Understanding	UND	TXO	SPO	FCO	GSO	SPC							
Translation	TRN	TXO	SPO	FCO	GSO	LNG							
Display	DIS	TXO	SPO	FCO	GSO								

The JSON file provides the formal specification of MPAI-HMC Profiles.

The Regular Expression can be interpreted by factoring it into its component rules:
^(ALL|NUL)([+-](AVS|UND|TRN|EPS|DIS)|
@AVS#(TXO|SPO|AUO|VIO|PAF(#(AVA|LNG|NNM|EPS|SPC|(TXO|SPO)#([a-
z]{3})(<->|->)([a-z]{3})))?)|
@UND#(TXO|SPO|FCO|GSO|SPC)|
@TRN#(TXO|SPO|FCO|GSO)#([a-z]{3})(<->|->)([a-z]{3})|
@EPS#(TXO|PSO|FCO|GSO)|
@DIS#(TXO|PSO|FCO|GSO))+$
An additional rule not specified by the Regular Expression is that ALL be always followed by "-" and NUL be always followed by "+".