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

MPAI Technical Specification

Multimodal Conversation
MPAI-MMC

V2.4

WARNING

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

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

Readers are invited to review Notices and Disclaimers.

Technical Specification
Multimodal Conversation (MPAI-MMC) V2.4

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.

Today, the MPAI organisation rests 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)* specifies an environment enabling initialisation, dynamic configuration, and control of AIWs in the standard AI Framework environment depicted in Figure 1. An AI Framework can execute AI applications called AI Workflows (AIW). An AIW includes interconnected AI Modules (AIM). MPAI-AIF supports small- and large-scale high-performance components and promotes solutions with improved explainability.

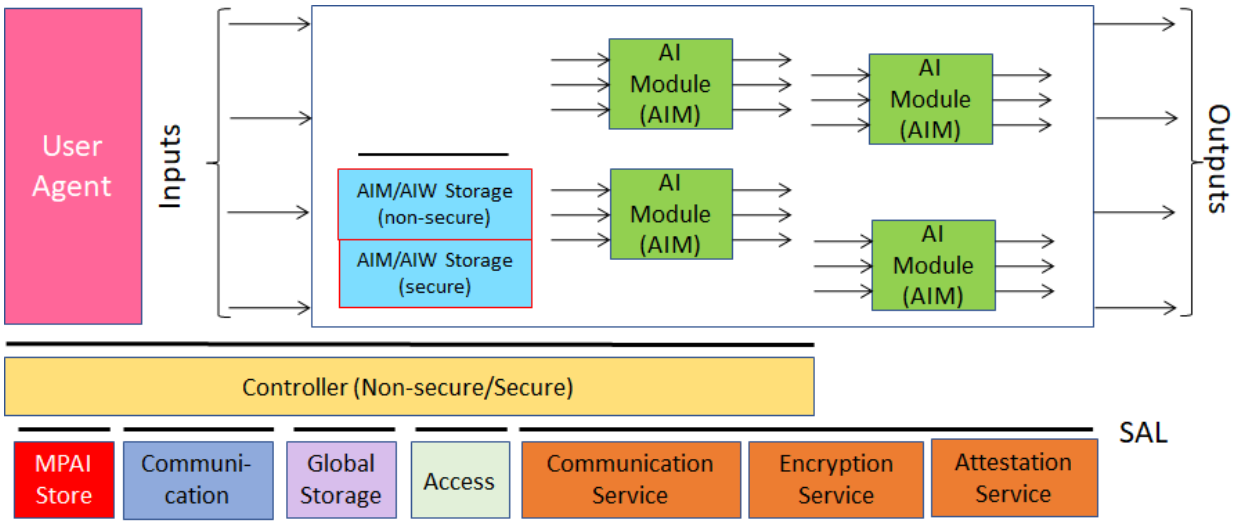

Figure 1 – The AI Framework (MPAI-AIF) V2 Reference Model

3. ***Technical Specification: Data Types, Formats, and Attributes (MPAI-TFA) V1.0*** specifies Qualifiers, a type of metadata supporting the operation of AIMs receiving data from other AIMs. 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, and Visual Data exchanged by AIWs and AIMs should be interpreted as being composed of Content (Text, Speech, Audio, and Visual as appropriate) and associated Qualifiers. The specifications of most MPAI Data Types reflect this point.
4. ***Technical Specification: Governance of the MPAI Ecosystem (MPAI-GME) V1.1*** defines the following elements:
5. Standards, i.e., the ensemble of Technical Specifications, Reference Software, Conformance Testing, and Performance Assessment.
6. Developers of MPAI-specified AIMs and Integrators of MPAI-specified AIWS (Implementers).
7. MPAI Store in charge of making AIMs and AIWs submitted by Implementers available to Integrators and End Users.
8. Performance Assessors, independent entities assessing the performance of implementations in terms of Reliability, Replicability, Robustness, and Fairness.
9. End Users.

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

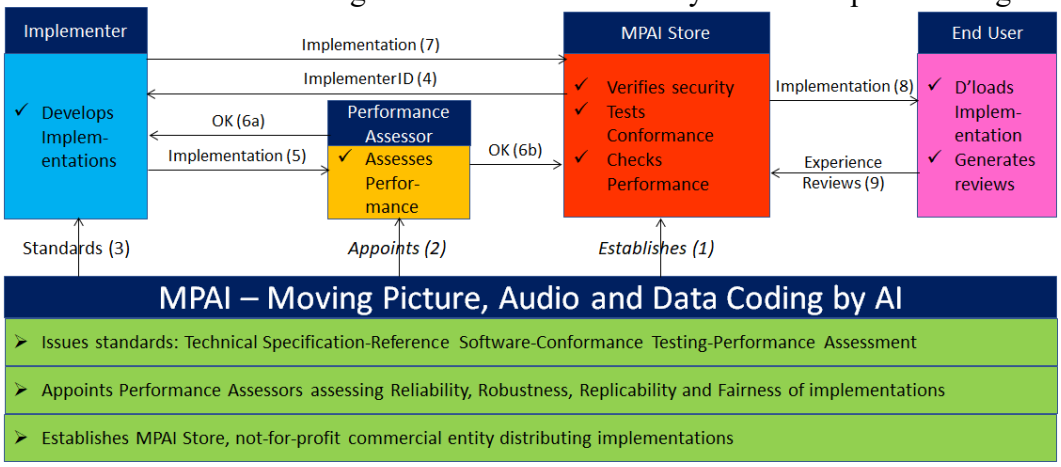

Figure 2 – The MPAI Ecosystem

2 Introduction (Informative)

From the moment a human built the first machine, there was a need to "communicate" with it. In the past, humans used more primitive machines to communicated by touch, later by characters and then with speech and even visual means. Then, with the appearance of more complex machines, the need for more sophisticated communication methods arose. Today, as personal devices become more pervasive, and the use of information and other online services become ubiquitous, the trend for human to communicate with machine is to be more direct and even "personal".

The ability of Artificial Intelligence to learn from interactions with humans gives machines the ability to improve their "conversational" capabilities by better understanding the meaning of what a human types or says and by providing more pertinent responses. If properly trained, machines can also learn to understand additional or hidden meanings of a sentence by analysing a human's text, speech, or gestures. Machines can also be made to develop and rely on "internal statuses" comparable to those driving the attitudes of conversing humans. and provide responses – in text, speech, and gestures – that are more human-like and richer in content.

***T*echnical Specification: Multimodal Conversation (MPAI-MMC) V2.4** has been developed by MPAI in pursuit of the following policies:

1. Be friendly to the AI context but, to the extent possible, agnostic to the technology – AI or Data Processing – used in an implementation.
2. Be attractive to different industries, end users, and regulators.
3. Address three levels of standardisation any of which an implementer can freely decide to adopt:
    1. Data types, i.e., the data exchanged by systems.
    2. Components (called AI Modules - AIM).
    3. Connections of components (called AI Workflows - AIW).
4. Specify the data exchanged by components with a semantic that is clear to the extent possible.

The MPAI-MMC V2 Technical Specification will be accompanied by the Reference Software, Conformance Testing, and Performance Assessment Specifications. Conformance Testing specifies methods enabling users to ascertain whether a data type generated by an AIM, an AIM, or an AIW conform with this Technical Specification.

The **MPAI-MMC V2.4** Technical Specification provides the technologies supporting the implementation of a subset or the totality of the possibilities envisaged by this Introduction:

1. It is organised by Use Cases, such as Conversation with Personal Status, Multimodal Question Answering, and Unidirectional Speech Translation, corresponding to AI Workflows.
2. Each Use Case provides:
    1. The functions.
    2. The Input/Output Data of the AIW implementing it.
    3. The Reference Model specifying the AIM topology.
    4. The AIMs specified in terms of functions performed and Input/Output Data.

In all Chapters and Sections, Terms beginning with a capital letter are defined in *Table 1* if they are specific to this Technical Specification. and in *Table 2* if they are common to all MPAI Technical Specifications. All MPAI-defined Terms are accessible online.

All Chapters, Sections, and Annexes are Normative unless they are labelled as Informative.

3 Scope

Technical Specification: Multimodal Conversation (MPAI-MMC) V2.4, in the following also called MPAI-MMC V2.4 or simply MPAI-MMC, specifies:

1. **Data Types** for use by MPAI-MMC V2.4 and other MPAI Technical Specifications.
2. **AI Modules** enabling analysis of text, speech, and other non-verbal components used in human-machine and machine-machine conversation applications.
3. **AI Workflows** implementing Use Cases that use AI Modules and Data Types from MPAI-MMC and other MPAI Technical Specifications to provide recognised applications in the Multimodal Conversation domain.

The Use Cases includes in this Technical Specification are:

1. *Answer to Multimodal Question* (MMC-AMQ) providing a text or speech answer to a text or speech question and an image.
2. *Conversation About a Scene* (MMC-CAS) where a human converses with a machine pointing at the objects scattered in a room and displaying Personal Status in their speech, face, and gestures while the machine responds displaying its Personal Status in speech, face, and gesture.
3. *Conversation with Personal Status* (MMC-CPS), enabling conversation and question answering with a machine able to extract the inner state of the entity it is conversing with and showing itself as a speaking digital human able to express a Personal Status. By adding or removing minor components to this general Use Case, five Use Cases are spawned:
4. *Conversation with Emotion* (MMC-CWE), enabling audio-visual conversation with a machine impersonated by a synthetic voice and an animated face.
5. *Human-Connected Autonomous Vehicle Interaction* (MMC-HCI) where humans converse with a machine displaying Personal Status after having been properly identified by the machine with their speech and face in outdoor and indoor conditions while the machine responds displaying its Personal Status in speech, face, and gesture.
6. *Multimodal Question Answering* (MQA), enabling request for information about a displayed object.
7. *Text and Speech Translation* (MMC-TST) supporting a variety of text and speech translation applications where users can specify whether speech or text is used as input and, if it is speech, whether their speech features are preserved in the interpreted speech.
8. *Virtual Meeting Secretary* (MMC-VSV) where an avatar not representing a human in a virtual avatar-based video conference extracts Personal Status from Text, Speech, Face, and Gestures, displays a summary of what other avatars say, and receives and act on comments.

The Composite AI Module specified by MPAI-MMC V2.3 is *Personal Status Extraction* (MMC-PSE) that estimates the Personal Status conveyed by Text, Speech, Face, and Gesture – of an Entity, i.e., a real or digital human.

Note that:

1. Each AI Workflow implementing a Use Case normatively defines:
    - The Functions of the AIW implementing it and of the AIMs.
    - The Connections between and among the AIMs
    - The Semantics and the Formats of the input and output data of the AIW and the AIMs.
2. Each AI Module normatively defines:
    - The Functions of the Composite AIM implementing it and of the AIMs.
    - The Connections between and among the AIMs
    - The Semantics and the Formats of the input and output data of the AIW and the AIMs.
3. Each Data Type:
    - Specifies a JSON Schema of the Data Type.
    - May include a reference to one of more Qualifiers.

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

1. An *AIW*, shall:
    1. Perform the AIW function specified in the appropriate Section of Chapter 5.

    2. All AIMs, their topology and connections should conform with the AIW Architecture specified in the appropriate Section of Chapter 5.
    3. The AIW and AIM input and output data should have the formats specified in the appropriate Sections of Chapter 7.
2. An *AIM*, shall:
    1. Perform the functions specified by the appropriate Section of Chapter 5 or 6.
    2. Receive and produce the data specified in the appropriate Section of Chapter 7.
    3. A data *Format*, the data shall have the format specified in Chapter 7.
3. A *Data Type*:
    1. Shall have a format conforming with the JSON Schema of the Data Type.
    2. May have Qualifier(s) referenced in the JSON Schema

Implementers of this Technical Specification should note that:

1. The Reference Software of this Technical Specification may be to develop Implementations.
2. The Conformance Testing specification may be used to test the conformity of an Implementation to this Standard.
3. The level of Performance of an Implementation may be assessed based on the Performance Assessment specification of this Standard.

Users should consider Notices and Disclaimers.

MPAI-MMC V2.4 has been developed by the MPAI Multimodal Conversation Development Committee (MM-DC). MPAI expects to produce future MPAI-MMC Versions extending the scope of the Use Cases and/or add new Use Cases supported by existing of new AI Modules and Data Types within the scope of Multimodal Conversation.

4 Definitions

Capitalised Terms have the meaning defined in *Table 1*. All MPAI-defined Terms are available online.

Lower case Terms have the meaning commonly defined for the context in which they are used. For instance, *Table 1* defines *Object* and *Scene* but does not define *object* and *scene*.

A dash "-" preceding a Term in *Table 1* indicates the following readings according to the font:

1. Normal font: the Term in the table without a dash and preceding the one with a dash should be read before that Term. For example, "Avatar" and "- Model" will yield "Avatar Model."
2. *Italic* font: the Term in *Table 1* without a dash and preceding the one with a dash should be read after that Term. For example, "Avatar" and "- Portable" will yield "Portable Avatar."

Table 1 – Table of terms and definitions

Term	Definition
Attitude	
- *Social*	The coded representation of the internal state related to the way a human or avatar intends to position vis-à-vis the Environment or subsets of it, e.g., "Respectful", "Confrontational", "Soothing".
- *Spatial*	Position and Orientation and their velocities and accelerations of an Audio and Visual Object in a Virtual Environment.
Audio	Digital representation of an analogue audio signal sampled at a frequency between 8-192 kHz with a number of bits/sample between 8 and 32, and non-linear and linear quantisation.
- Object	Coded representation of Audio information with its metadata. An Audio Object can be a combination of Audio Objects.

- Scene	The Audio Objects of an Environment with Object location metadata.
Audio-Visual Object	Coded representation of Audio-Visual information with its metadata. An Audio-Visual Object can be a combination of Audio-Visual Objects.
Audio-Visual Scene	(AV Scene) The Audio-Visual Objects of an Environment with Object location metadata.
Avatar	An animated 3D object representing a real or fictitious person in a Virtual Space.
- Model	An inanimate avatar exposing interfaces enabling animation.
Cognitive State	The coded representation of the internal state reflecting the way a human or avatar understands the Environment, such as “Confused”, “Dubious”, “Convinced”.
Colour (of speech)	The timber of an identifiable voice independent of a current Personal Status and language.
Connected Autonomous Vehicle	A vehicle able to autonomously reach an assigned geographical position by: 1. Understanding human utterances. 2. Planning a route. 3. Sensing and interpreting the Environment. 4. Exchanging information with other CAV. 5. Acting on the CAV’s motion actuation subsystem.
Context	Additional information about a communication emitted by an Entity, such as language, culture etc..
Data	Information in digital form.
- Format	The standard digital representation of Data.
- Type	An instance of Data with a specific Data Format.
Descriptor	Coded representation of text, audio, speech, or visual feature.
Digital Representation	Data corresponding to and representing a real entity.
Emotion	The coded representation of the internal state resulting from the interaction of a human or avatar with the Environment or subsets of it, such as “Angry”, “Sad”, “Determined”.
Entity	A real or Digital Human
Environment	A Virtual Space containing a Scene.
Face	The portion of a 2D or 3D digital representation corresponding to the face of a human.
Factor	One of Emotion, Cognitive State and Attitude.
Gesture	A movement of the body or part of it, such as the head, arm, hand, and finger, often a complement to a vocal utterance.
Grade	The intensity of a Factor.
Human	A human being in a real space.
- *Digital*	A Digitised or a Virtual Human in a Virtual Space.
- *Digitised*	An Object in a Virtual Space that has the appearance of a specific human when rendered.

- *Virtual*	An Object in a Virtual Space created by a computer that has a human appearance when rendered but is not a Digitised Human.
Identifier	The label uniquely associated with a human or an avatar or an object.
Instance	An element of a set of entities – Objects, users etc. – belonging to some levels in a hierarchical classification (taxonomy).
Intention	The result of analysis of the goal of an input question.
Manifestation	The manner of showing the Personal Status, or a subset of it, in any one of Speech, Face, and Gesture.
Meaning	Information extracted from Text such as syntactic and semantic information, Personal Status, and other information, such as an Object Identifier.
Modality	One of Text, Speech, Face, or Gesture.
Object Descriptors	Attribute of the coded representation of an object in a Scene, including its Spatial Attitude.
Orientation	The set of the 3 roll, pitch, yaw angles indicating the rotation around the principal axis (x) of an Object, its y axis having an angle of 90° counterclockwise (right-to-left) with the x axis and its z axis pointing up toward the viewer.
Personal Status	The ensemble of information internal to a person, including Emotion, Cognitive State, and Attitude.
Portable Avatar	A Data Type representing an Avatar and its Context.
Pitch	The fundamental frequency of Speech. Pitch is the attribute that makes it possible to judge sounds as "higher" and "lower."
Point of View	The Spatial Attitude of a human or avatar looking at an Environment.
Position	The 3 coordinates (x,y,z) of a representative point of an object in the Real and Virtual Space.
Refined Text	The Text resulting from the analysis of the Text produced by Automatic Speech Recognition made by Natural Language Understanding.
Scene	A structured composition of Objects.
Speech	Digital representation of analogue speech sampled at a frequency between 8 kHz and 96 kHz with a number of bits/sample of 8, 16 and 24, and non-linear and linear quantisation.
- Features	Aspects of a speech segment that enable its description and reproduction, e.g., degree of vocal tension, Pitch, etc., and that can be automatically recognised and extracted for speech synthesis or other related purposes.
- Rate	The number of Speech Units per second.
- Unit	Phoneme, syllable, or word as a segment of Speech.
Summary	An abridged outline of the content of the utterance(s) of one or more Users possibly including their Personal Statuses.
Text	A sequence of characters drawn from a finite alphabet.
Visual Object	Coded representation of Visual information with its metadata. A Video Object can be a combination of Video Objects.
Vocal Gesture	Utterance, such as cough, laugh, hesitation, etc. Lexical elements are excluded.

5 References

5.1 Normative References

This standard normatively references the following documents, both from MPAI and other standards organisations. MPAI standards are publicly available at https://mpai.community/standards/resources/.

1. MPAI; Technical Specification: Governance of the MPAI Ecosystem (MPAI-GME) V2.0.
2. MPAI; Technical Specification: Artificial Intelligence Framework (MPAI-AIF) V2.2.
3. MPAI; Technical Specification: Context-based Audio Enhancement (MPAI-CAE) - Use Cases (CAE-USC) V2.4.
4. MPAI; Technical Specification: MPAI Metaverse Model (MPAI-MMM) – Technologies (MMM-TEC) V2.0.
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 Specifications: Profiles (MPAI-PRF) - AI-Modules (PRF-AIM) V1.1.
8. MPAI; Technical Specification: Data Types, Formats, & Attributes (MPAI-TFA) V1-4.

5.2 Informative References

The references provided here are for information purpose.

9. MPAI; The MPAI Statutes.
10. MPAI; The MPAI Patent Policy.
11. MPAI; Framework Licence of the Multimodal Conversation Technical Specification (MPAI-MMC) V1; https://mpai.community/standards/mpai-mmc/framework-licence/mpai-mmc-v1-framework-licence/.
12. MPAI; Framework Licence of the Multimodal Conversation Technical Specification (MPAI-MMC) V2; https://mpai.community/standards/mpai-mmc/call-for-technologies/mpai-mmc-v2-call-for-technologies/.
13. Ekman, Paul (1999), "Basic Emotions", in Dalgleish, T; Power, M (eds.), Handbook of Cognition and Emotion (PDF), Sussex, UK: John Wiley & Sons.
14. Emotion Markup Language (EmotionML) 1.0; https://www.w3.org/TR/2010/WD-emotionml-20100729/diffmarked.html.
15. Hobbs J.R., Gordon A.S. (2011) The Deep Lexical Semantics of Emotions. In: Ahmad K. (eds) Affective Computing and Sentiment Analysis. Text, Speech, and Language Technology, vol 45. Springer, Dordrecht, https://people.ict.usc.edu/~gordon/publications/EMOT08.PDF and https://www.researchgate.net/publication/227251103_The_Deep_Lexical_Semantics_of_Emotions.

6 AI Workflows

6.1 Technical Specification

Technical Specification: Multimodal Conversation (MPAI-MMC) V2.4 assumes that Workflow implementations will be based on *Technical Specification: AI Framework (MPAI-AIF) V2.1*. specifying an AI Framework (AIF) where AI Workflows (AIW) composed of interconnected AI Modules (AIM) are executed.

Table 1 provides the full list of AIWs specified by MPAI-MMC V2.4 with links to the pages dedicated to each AI Workflow which includes its function, reference model, Input/Output Data,

Functions of AIMs, Input/Output Data of AIMs, and links to the AIW-related AIW, AIMs, and JSON metadata.
All MPAI-MMC V2.3 specified AI-Workflows are superseded by those specified by MPAI-MMC V2.4. MPAI-MMC V2.3 specification can still be used if their version is explicitly indicated.

Table 1 - AIWs of MPAI-MMC V2.4

Acronym.	Title	JSON	Acronym	Title	JSON
MMC-AMQ	Answer to Multimodal Question	X	MMC-HCI	Human-CAV Interaction	X
MMC-CAS	Conversation About a Scene	X	MMC-MQA	Multimodal Question Answering	X
MMC-CPS	Conversation with Personal Status	X	MMC-TST	Text and Speech Translation	X
MMC-CWE	Conversation with Emotion	X	MMC-VMS	Virtual Meeting Secretary	X

6.1.1 Answer to Multimodal Question

6.1.1.1 Functions

The Answer to Multimodal Question MMC-AMQ) receives a question expressed as a Text Object or a Speech Object and an Image and provides Text and/or Speech giving information in response to the question.

6.1.1.2 Reference Model

Figure 1 specifies the Answer to Multimodal Question (MMC-AMQ) Reference Model including the input/output data, the AIMs, and the data exchanged between and among the AIMs.

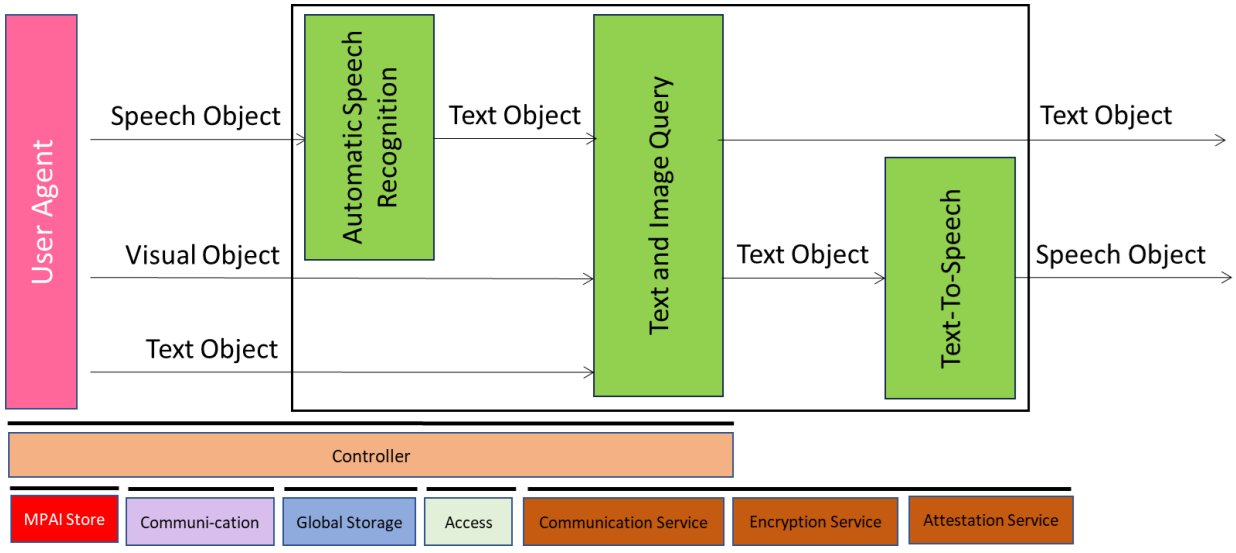

Figure 1 – Reference Model of Answer to Multimodal Question (MMC-AMQ)

The operation of Answer to Multimodal Question (MMC-AMQ) develops in the following way:

1. A user provides
    1. Text Object or Speech Object
    2. An Image

2. The machine provides the answer expressed as Text Object and/or Speech Object.

6.1.1.3 I/O Data

The input and output data of the Answer to Multimodal Question (MMC-AMQ) Use Case are:

Table 1 – I/O Data of Multimodal Question Answering

Input	Descriptions
Text Object	Text typed by the human as a replacement for Input Speech.
Image	Image about which a question is asked.
Speech Object	Speech question to the Machine.
Output	**Descriptions**
Machine Text	The Text generated by Machine in response to human input.
Machine Speech	The Speech generated by Machine in response to human input.

6.1.1.4 Functions of AI Modules

Table 2 provides the functions of the Answer to Multimodal Question (MMC-AMQ) Use Case.

Table 2 – Functions of AI Modules of Multimodal Question Answering (MMC-AMQ)

AIM	Function
Automatic Speech Recognition	Recognises Speech.
Text and Image Query	Produces Text response to the query.
Text-to-Speech	Synthesises Speech from Text.

6.1.1.5 I/O Data of AI Modules

The AI Modules of Multimodal Question Answering (MMC-AMQ) are given in *Table 3*.

Table 3 – AI Modules of Multimodal Question Answering (MMC-AMQ)

AIM	Receives	Produces
Automatic Speech Recognition	Speech Object	Recognised Text
Text and Image Query	1. Input or Recognised Text 2. Image Visual Object	Machine Text
Text-to-Speech	Machine Text	Machine Speech

6.1.1.6 AIW, AIMs, and JSON Metadata

Table 4 provides the links to the AIW and AIM specifications and to the JSON syntaxes. AIMs/1 indicates that the column contains Composite AIMs and AIMs indicates that the column contains their Basic AIMs.

Table 4 – AIW, AIMs, and JSON Metadata

AIW	AIMs	Name	JSON

MMC-AMQ		Answer to Multimodal Question	X
	MMC-ASR	Automatic Speech Recognition	X
	MMC-TIQ	Text and Image Query	X
	MMC-TTS	Text-to-Speech	X

6.1.1.7 Reference Software

6.1.1.7.1 Disclaimers

1. This MMM-AMQ 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-AMQ, 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.

6.1.1.7.2 Guide to the AMQ code

Use of this AI Workflow is for developers who are familiar with Python and downloading models from HuggingFace,
A wrapper for three model is provided the Whisper (ASR), BLIP (TIQ), and speech5 (TTS):

1. Manages input files and parameters: Speech Object, Visual Object, Text Object
2. Executes the AIW to perform the Answer to Multimodal Question on each individual pair of Speech/Text and Visual Object.
3. Outputs the answer as Speech Object and Text Object.

The OSD-AQM Reference Software is found at the NNW gitlab site. It contains:

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

6.1.1.7.3 Acknowledgements

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

6.1.1.7.4 Conformance Testing

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

Receives	Text Object	Shall validate against Text Object Schema. Text Data shall conform with Text Qualifier.
	Image Visual Object	Shall validate against Visual Object Schema. Text Data shall conform with Visual Qualifier.

	Speech Object	Shall validate against Speech Object Schema. Text Data shall conform with Speech Qualifier.
Produces	Machine Text	Shall validate against Text Object Schema. Text Data shall conform with Text Qualifier.
	Machine Speech	Shall validate against Speech Object Schema. Text Data shall conform with Speech Qualifier.

Table 6 provides an example of MMC-AMQ AIM conformance testing.

Table 6 – An example MMC-AMQ AIM conformance testing

Input Data	**Data Type**	**Input Conformance Testing Data**
Text Object	Unicode	All input Text files to be drawn from Text Files.
Speech Object	.wav	All input Speech files to be drawn from Audio Files.
Input Image	JPEG	All input Image files to be drawn from Images.
Output Data	**Data Type**	**Input Conformance Testing Data**
Machine Text	Unicode	All Text files produced shall conform with Text files.
Machine Speech	.wav	All Speech files produced shall conform with Speech files.

6.1.2 Conversation About a Scene

This Use Case addresses the case of a human holding a conversation with a Machine:

1. The human converses with the Machine indicating the object in the Environment s/he wishes to talk to or ask questions about it using Speech, Face, and Gesture.
2. The Machine
    - Sees and hears an Environment containing a speaking human and some scattered objects.
    - Recognises the human's Speech and obtains the human's Personal Status by capturing Speech, Face, and Gesture.
    - Understands which object the human is referring to and generates an avatar that:
        - Utters Speech conveying a synthetic Personal Status that is relevant to the human's Personal Status as shown by his/her Speech, Face, and Gesture, and
        - Displays a face conveying a Personal Status that is relevant to the human's Personal Status and to the response the Machine intends to make.
    - Renders the Scene that it perceives from a human-selected Point of View. The objects in the scene are labelled with the Machine's understanding of their semantics so that the human can understand how the Machine sees the Environment.

6.1.2.1 Reference Model

Figure 1 gives the Conversation About a Scene Reference Model including the input/output data, the AIMs, and the data exchanged between and among the AIMs.

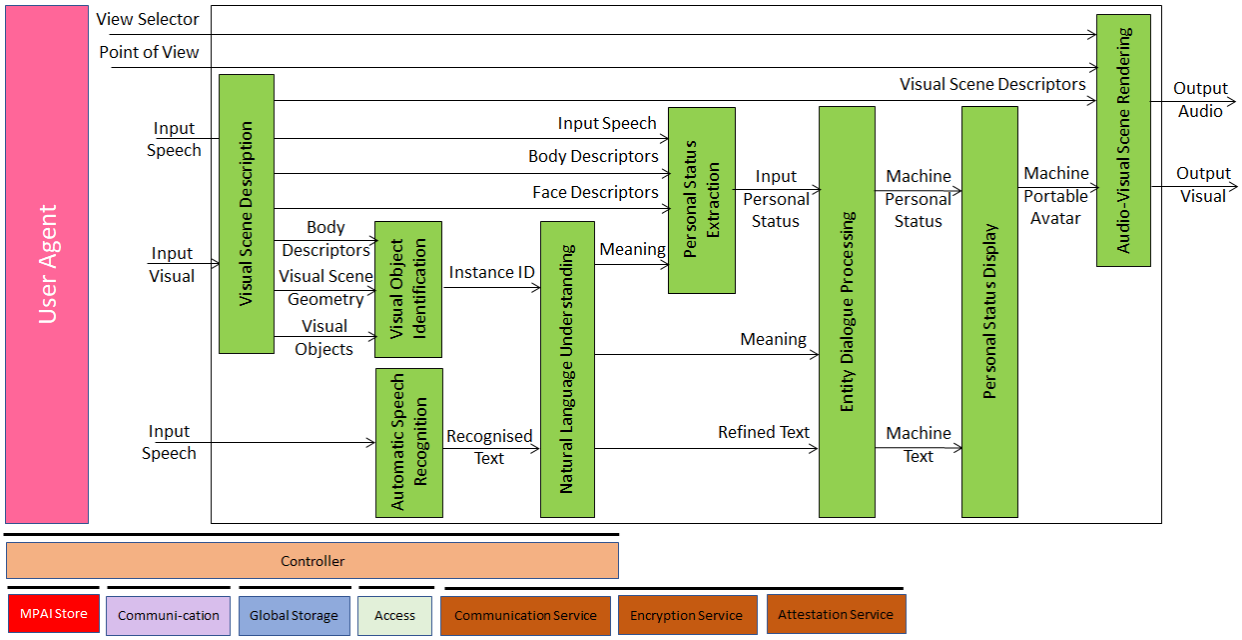

Figure 1 – Reference Model of Conversation About a Scene (MMC-CAS) AIM

The Machine operates according to the following workflow:

1. Visual Scene Description produces Body Descriptors, Visual Scene Geometry and Visual Objects from Input Visual.
2. Automatic Speech Recognition produces Recognised Text from Input Speech.
3. Visual Object Identification produces Visual Object Instance ID from Visual Objects, Body Descriptors, and Visual Scene Geometry.
4. Natural Language Understanding produces Meaning and Refined Text from Recognised Text and Visual Object ID.
5. Personal Status Extraction produces Input Personal Status from Meaning, Input Speech, Face Descriptors, and Body Descriptors.
6. Entity Dialogue Processing produces Machine Text and Machine Personal Status from Input Personal Status, Meaning, and Refined Text.
7. Personal Status Display produces Machine Portable Avatar from Machine Text, and Machine Personal Status.
8. Audio-Visual Scene Rendering renders the Audio-Visual Scene
    1. Described by the Visual Scene Descriptors.
    2. Integrated by the Machine's Portable Avatar information depending on View Selector.
    3. As seen from the human-selected Point of View.

6.1.2.2 I/O Data

Table 1 gives the input/output data of Conversation About a Scene.

Table 1 – I/O data of Conversation About a Scene

Input data	From	Description
View Selector	Human	Selects whether Machine is rendered in the scene
Input Visual	Camera	Points to human and scene.
Input Speech	Microphone	Speech of human.

Point of View	Human	The point of view of the Audio-Visual Scene displayed by Audio-Visual Scene Rendering.
Output data	**To**	**Descriptions**
Output Visual	Human	Rendering of the Visual Scene containing labelled objects, human, and Machine depending on View Selector as perceived by Machine and seen from the Point of View.
Output Speech	Human	Speech of Portable Avatar produced by Machine.

6.1.2.3 Functions of AI Modules

Table 2 provides the functions of the Conversation About a Scene Use Case.

Table 2 – Functions of AI Modules of Conversation About a Scene

AIM	Functions
Visual Scene Description	1. Receives Input Visual 2. Provides Visual Objects and Visual Scene Geometry.
Visual Object Identification	1. Receives Body Descriptors and non-human Visual Objects 2. Provides the Instance ID of the Visual Object indicated by the human.
Automatic Speech Recognition	1. Receives Input Speech 2. Provides Recognised Text.
Natural Language Understanding	1. Receives Instance ID and Recognised Text 2. Refines Text and extracts Meaning.
Personal Status Extraction	1. Receives Input Speech, Body Descriptors, Face Descriptors, and Meaning. 2. Provides Personal Status.
Entity Dialogue Processing	1. Receives Refined Text and Personal Status. 2. Produces Machine's Text and Personal Status.
Personal Status Display	1. Receives Machine's Personal Status and Text. 2. Provides Machine Portable Avatar.
Audio-Visual Scene Rendering	1. Receives the Descriptors of the Visual Scene perceived by Machine including the Portable Avatar of the Personal Status Display. 2. Renders the Audio-Visual Scene from the Point of View selected by human.

6.1.2.4 I/O Data of AI Modules

Table 3 gives the list of AIMs with their I/O Data.

Table 3 – AI Modules of Conversation About a Scene

AIM	Receives	Produces
Visual Scene Description	Input Visual	1. Visual Scene Descriptors 2. Body Descriptors 3. Face Descriptors

		4. Visual Scene Geometry 5. Visual Objects
Visual Object Identification	1. Body Visual Object 2. Visual Objects 3. Visual Scene Geometry	1. Visual Object Instance Identifier
Automatic Speech Recognition	1. Input Speech	1. Recognised Text
Natural Language Understanding	1. Recognised Text 2. Visual Object Instance Identifier	1. Meaning 2. Refined Text
Personal Status Extraction	1. Body Visual Object 2. Face Visual Object 3. Input Speech 4. Meaning	1. Personal Status
Entity Dialogue Processing	1. Personal Status 2. Meaning 3. Visual Object ID 4. Refined Text	1. Machine Personal Status
Personal Status Display	1. Machine Text 2. Machine Personal Status	1. Machine Portable Avatar
Audio-Visual Scene Rendering	1. Visual Scene Descriptors 2. Point of View	1. Output Speech 2. Output Visual

6.1.2.5 AIW, AIMs, and JSON Metadata and AIMs

Table 4 provides the links to the AIW and AIM specifications and to the JSON syntaxes. AIMs/1 indicates that the column contains Composite AIMs and AIMs/2 indicates that the column contains their Basic AIMs.

Table 4 – AIW, AIMs, and JSON Metadata

AIW	AIMs/1	AIMs/2	Name	JSON
MMC-CAS			Conversation About a Scene	X
	OSD-VSD		Visual Scene Description	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
	MMC-ASR		Automatic Speech Recognition	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-EDP		Entity Dialogue Processing	X
	OSD-PSD		Personal Status Display	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
	PAF-AVR		Audio-Visual Scene Rendering	X

6.1.2.6 Reference Software

6.1.2.7 Conformance Testing

Table 5 provides the Conformance Testing Method for MMC-CAS 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 MMC-CAS AIM

Receives	View Selector	Shall validate against Selector Schema.
	Input Visual	Shall validate against Visual Object Schema. Speech Data shall conform with Visual Qualifier.
	Input Speech	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.
	Point of View	Shall validate against Point of View Schema.
Produces	Output Visual	Shall validate against Visual Object Schema. Speech Data shall conform with Visual Qualifier.
	Output Speech	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.

6.1.3 Conversation with Personal Status

6.1.3.1 Functions

When humans have a conversation with other humans, they use speech and, in constrained cases, text. Their interlocutors perceive speech and/or text supplemented by visual information related to the speaker's face and gesture of a conversing human. Text, speech, face, and gesture may convey information about the internal state of the speaker that MPAI calls Personal Status. Therefore, handling of Personal Status information in a human-machine conversation and, in the future, even machine-machine conversation, is a key feature of a machine trying to understand what the speakers' utterances mean because Personal Status recognition can improve understanding of the speaker's utterance and help a machine produce better replies.
Conversation with Personal Status (MMC-CPS) is a general Use Case of an Entity – a real human or Digital Human – conversing with and asking questions to a machine. The machine captures and understands Text and Speech, extracts Personal Status from the Text, Speech, Face, and Gesture Factors, fuses the Factors' Personal Statuses into an estimated Personal Status of the Entity to achieve a better understanding of the context in which the Entity utters Speech.

6.1.3.2 Reference Model

Figure 1 gives the Conversation with Personal Status Reference Model including the input/output data, the AIMs, and the data exchanged between and among the AIMs.
The operation of the Conversation with Personal Status Use Case develops as follows:

1. Input Selector is used to inform the machine whether the human employs Text or Speech in conversation with the machine.
2. Visual Scene Description extracts the Scene Geometry, the Visual Objects and the Face and Body Descriptors of humans in the Scene.
3. Audio Scene Description extracts the Scene Geometry, and the Speech Objects in the Scene.
4. Visual Object Identification assigns an Identifier to each Visual Object indicated by a human.
5. Audio-Visual Alignment uses the Audio Scene Description and Visual Scene Description to assign unique Identifiers to Audio, Visual, and Audio-Visual Objects.
6. Automatic Speech Recognition recognises Speech utterances.
7. Natural Language Understanding refines Text and extracts Meaning.
8. Personal Status Extraction extracts a human's Personal Status.
9. Entity Dialogue Processing produces the machine's response and its Personal Status.
10. Personal Status Display produces a speaking Avatar expressing Personal Status.
11. Audio-Visual Scene Rendering produces Machine Text, Speech, and Visual.

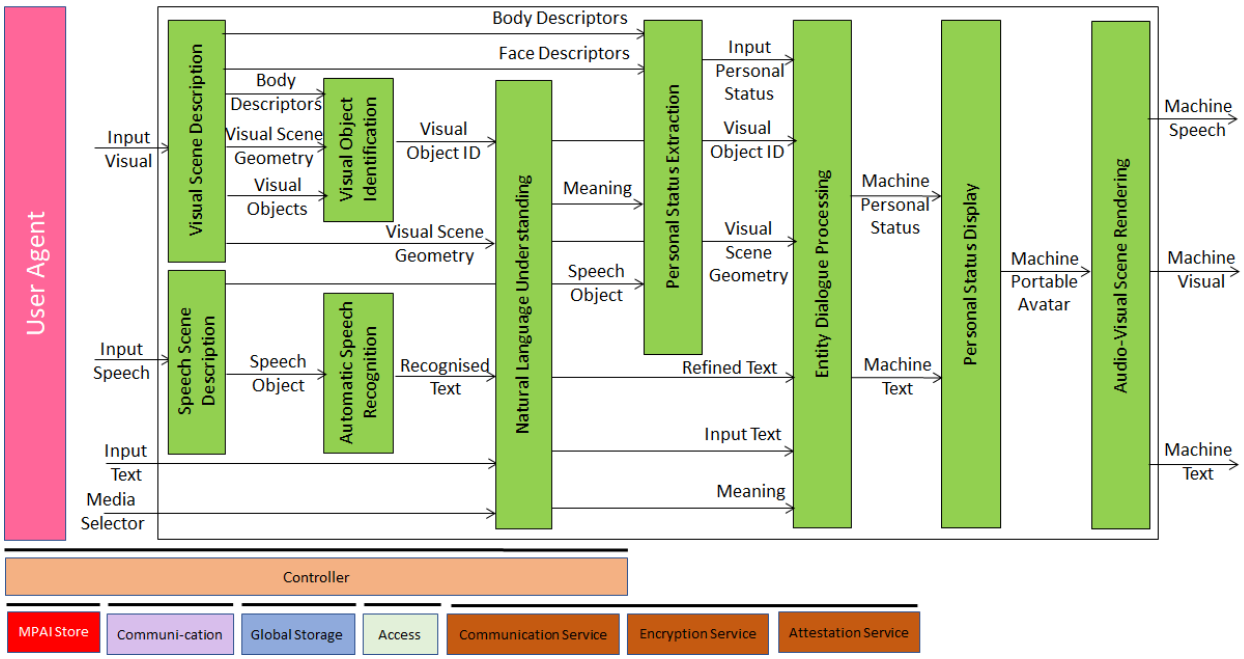

Figure 1 – Reference Model of Conversation with Personal Status

The operation of the Conversation with Personal Status Use Case develops as follows:

1. Selector is used to inform the machine whether the human employs Text or Speech in conversation with the machine.
2. Audio-Visual Scene Description extracts Audio Scene Geometry, Visual Scene Geometry, Audio Objects, Visual Objects, Face Descriptors and Body Descriptors of human in the Scene.
3. Visual Object Identification assigns an Identifier to each Visual Object indicated by a human.
4. Audio-Visual Alignment uses the Audio Scene Descriptors and Visual Scene Descriptors to assign unique Identifiers to Audio, Visual, and Audio-Visual Objects.
5. Automatic Speech Recognition recognises Speech utterances.
6. Natural Language Understanding refines Text and extracts Meaning.
7. Personal Status Extraction extracts the human's Personal Status.
8. Entity Dialogue Processing produces the machine's response as Text and Personal Status.
9. Personal Status Display produces a speaking Portable Avatar expressing Personal Status.
10. Audio-Visual Rendering produces Audio, Visual, and Text.

6.1.3.3 I/O Data

Table 1 gives the input and output data of the Conversation with Personal Status Use Case:

Table 1 – I/O Data of Conversation with Personal Status

Input	**Descriptions**
Input Text	Text typed by the human as additional information stream or as a replacement of the Speech.
Input Speech	Speech of the human having a conversation with the machine.
Input Visual	Visual information of the Face and Body of the human having a conversation with the machine.
Media Selector	Data determining the use of Speech vs Text.
Output	**Descriptions**

Output Text	Machine's Text
Output Speech	Machine's Audio (Speech)
Output Visual	Machine's Visual

6.1.3.4 Functions of AI Modules

Table 2 provides the functions of the Conversation with Personal Status Use Case.

Table 2 – Functions of AI Modules of Conversation with Personal Status

AIM	Function
Visual Scene Description	1. Receives Input Visual. 2. Provides Visual Objects and Visual Scene Geometry.
Speech Scene Description	1. Receives Input Speech. 2. Provides Speech Object.
Visual Object Identification	1. Receives Visual Scene Geometry, Body Descriptors, and Visual Objects. 2. Provides Visual Object Instance IDs.
Automatic Speech Recognition	1. Receives Speech Object. 2. Extracts Recognised Text.
Natural Language Understanding	1. Receives Recognised Text, Visual Object ID, and Visual Scene Geometry 2. Refines Text and extracts Meaning.
Personal Status Extraction	1. Receives Meaning, Refined Text, Body Descriptors, and Face Descriptors. 2. Extracts Personal Status.
Entity Dialogue Processing	1. Receives Refined Text, Personal Status, Visual Object ID, and Visual Scene Geometry. 2. Produces Machine's Text and Personal Status.
Personal Status Displays	1. Receives Machine Text and Personal Status. 2. Multiplexes Machine Text and Personal Status into Machine Portable Avatar.
Audio-Visual Scene Rendering	1. Receives Portable Avatar 2. Produces Machine Text, Machine Speech, and Machine Visual.

6.1.3.5 I/O Data of AI Modules

Table 3 provides the I/O Data of the AI Modules of the Conversation with Personal Status Use Case.

Table 3 – I/O Data of AI Modules of Conversation with Personal Status

AIM	Receives	Produces
Visual Scene Description	1. Input Visual	1. Face Descriptors 2. Body Descriptors 3. Audio-Visual Scene

		Descriptors 4. Visual Objects
Speech Scene Description	1. Input Speech	1. Speech Object
Visual Object Identification	1. Body Descriptors 2. Visual Scene Geometry 3. Visual Objects	1. Visual Object ID
Automatic Speech Recognition	1. Input Speech	1. Recognised Text
Natural Language Understanding	1. Visual Object ID 2. Input Speech 3. Recognised Text 4. Input Selector	1. Meaning 2. Refined Text
Personal Status Extraction	1. Face Descriptors 2. Body Descriptors 3. Meaning 4. Speech	1. Input Personal Status
Entity Dialogue Processing	1. Input Speech 2. Refined Speech 3. Input Personal Status 4. Input Selector	1. Machine Personal Status 2. Machine Speech
Personal Status Displays	1. Machine Speech 2. Machine Personal Status	1. Machine Portable Avatar
Audio-Visual Scene Rendering	1. Machine Portable Avatar	1. Machine Text 2. Machine Speech 3. Machine Visual

6.1.3.6 JSON Metadata

Table 4 provides the links to the AIW and AIM specifications and to the JSON syntaxes. AIMs/1 indicates that the column contains Composite AIMs and AIMs/2 indicates that the column contains their Basic AIMs.

Table 4 – Acronyms and URLs of JSON Metadata

AIW	AIMs/1	AIMs/2	Name	JSON
MMC-CPS			Conversation With Personal Status	X
	OSD-AVS		Audio-Visual Scene Description	X
	MMC-SSD		Speech Scene Description	X
	OSD-VSD		Visual Scene Description	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
	MMC-ASR		Automatic Speech Recognition	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-EDP		Entity Dialogue Processing	X
	PAF-PSD		Personal Status Display	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
	OSD-AVR		Audio-Visual Scene Rendering	X

6.1.3.7 Conformance Testing

Table 5 provides the Conformance Testing Method for MMC-CPS 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 MMC-CPS AIM

Receives	Input Text	Shall validate against Text Object Schema. Speech Data shall conform with Text Qualifier.
	Input Speech	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.
	Input Visual	Shall validate against Visual Object Schema. Speech Data shall conform with Visual Qualifier.
	Media Selector	Shall validate against Selector Schema.

Produces	Output Text	Shall validate against Text Object Schema. Speech Data shall conform with Text Qualifier.
	Output Speech	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.
	Output Visual	Shall validate against Visual Object Schema. Speech Data shall conform with Visual Qualifier.

6.1.4 Conversation with Emotion

6.1.4.1 Functions

In the Conversation with Emotion (MMC-CWE) Use Case, a machine responds to a human's textual and/or vocal utterance in a manner consistent with the human's utterance and emotional state, as detected from the human's text, speech, or face. The machine responds using text, synthetic speech, and a face whose lip movements are synchronised with the synthetic speech and the synthetic machine emotion.

6.1.4.2 Reference Model

Figure 1 gives the Reference Model of Conversation With Emotion including the input/output data, the AIMs, the AIM topology, and the data exchanged between and among the AIMs.

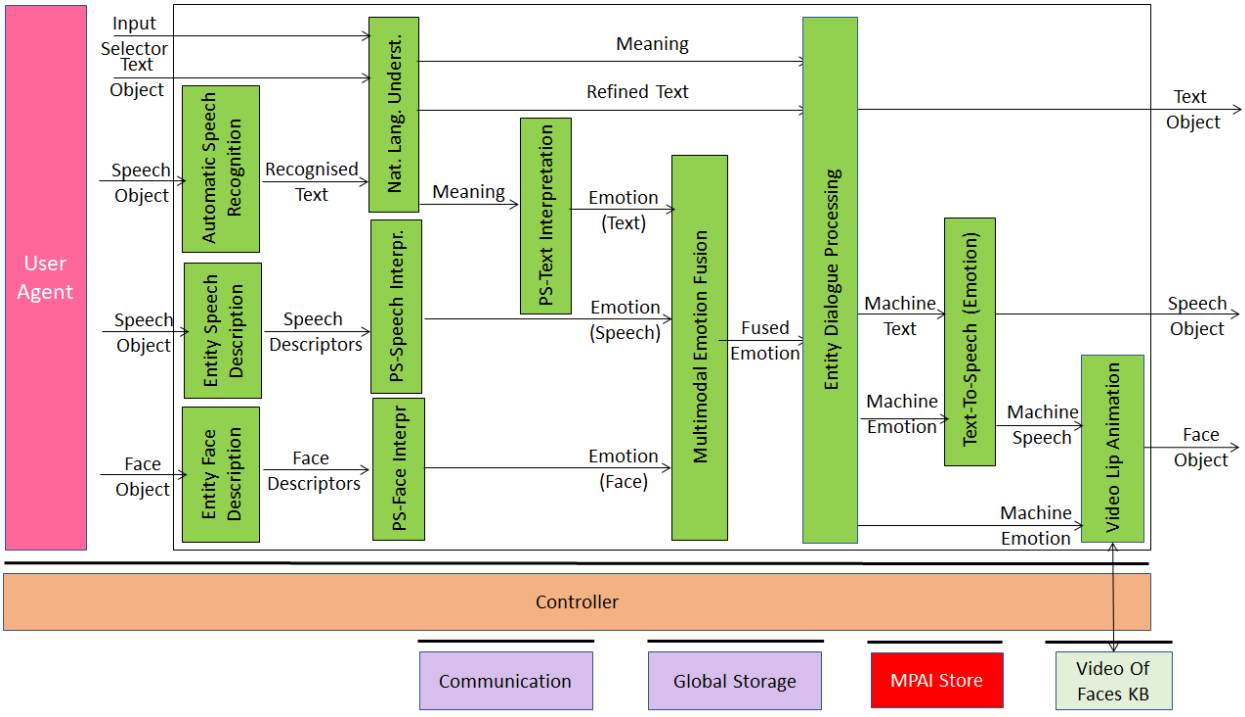

Figure 1 – Reference Model of Conversation With Emotion

The operation of Conversation with Emotion develops as follows:

1. Automatic Speech Recognition produces Recognised Text
2. Input Speech Description and PS-Face Interpretation produce Emotion (Speech).
3. Input Face Description and PS-Face Interpretation produce Emotion (Face).
4. Natural Language Understanding refines Recognised Text and produces Meaning.
5. Input Text Description and PS-Text Interpretation produce Emotion (Text).
6. Multimodal Emotion Fusion AIM fuses all Emotions into the Fused Emotion.

7. The Entity Dialogue Processing AIM produces a reply based on the Fused Emotion and Meaning.
8. The Text-To-Speech (Emotion) AIM produces Output Speech from Text with Emotion.
9. The Lips Animation AIM animates the lips of a Face drawn from the Video of Faces KB consistently with the Output Speech and the Output Emotion.

6.1.4.3 I/O Data

The input and output data of the Conversation with Emotion Use Case are:

Table 1 – I/O Data of Conversation with Emotion

Input	**Descriptions**
Input Selector	Data determining the use of Speech vs Text.
Text Object	Text typed by the human as additional information stream or as a replacement of the speech depending on the value of Input Selector.
Speech Object	Speech of the human having a conversation with the machine.
Face Visual Object	Visual information of the Face of the human having a conversation with the machine.
Output	**Descriptions**
Text Object	Text of the Speech produced by the Machine.
Speech Object	Synthetic Speech produced by the Machine.
Face Visual Object	Video of a Face whose lip movements are synchronised with the Output Speech and the synthetic machine emotion.

6.1.4.4 Functions of AI Modules

Table 2 provides the functions of the Conversation with Emotion AIMs.

Table 2 – Functions of AI Modules of Conversation with Emotion

AIM	**Function**
Automatic Speech Recognition	1. Receives Speech Object. 2. Produces Recognised Text.
Entity Speech Description	1. Receives Speech Object. 2. Produces Speech Descriptors
Entity Face Description	1. Receives Face Object. 2. Extracts Face Descriptors.
Natural Language Understanding	1. Receives Input Selector, Text Object, Recognised Text. 2. Produces Meaning (i.e., Text Descriptors), Refined Text.
PS-Text Interpretation	1. Receives Text Descriptors. 2. Provides the Emotion of the Text.
PS-Speech Interpretation	1. Receives Speech Descriptors. 2. Provides the Emotion of the Speech.

PS-Face Interpretation	1. Receives Face Descriptors. 2. Provides the Emotion of the Face.
Multimodal Emotion Fusion	1. Receives Emotion (Text), Emotion (Speech), Emotion (Face). 2. Provides human's Input Emotion by fusing Emotion (Text), Emotion (Speech), and Emotion (Video).
Entity Dialogue Processing	1. Receives Refined Text, Meaning, Input Emotion. 2. Analyses Meaning and Input Text or Refined Text, depending on the value of Input Selector. 3. Produces Machine Emotion and Machine Text.
Text-to-Speech	1. Receives Machine Text and Machine Emotion. 2. Produces Output Speech.
Video Lip Animation	1. Receives Machine Speech and Machine Emotion. 2. Animates the lips of a video obtained by querying the Video Faces KB, using the Output Emotion. 3. Produces Face Object with synchronised Speech Object (Machine Object).

6.1.4.5 I/O Data of AI Modules

The AI Modules of Conversation with Emotion perform the Functions specified in Table 21.

Table 3 – AI Modules of Conversation with Emotion

AIM	Receives	Produces
Automatic Speech Recognition	Speech Object	Recognised Text
Entity Speech Description	Speech Object	Speech Descriptors
Entity Face Description	Face Visual Object	Face Descriptors
Natural Language Understanding	Recognised Text	Refined Text Text Descriptors
PS-Text Interpretation	Text Descriptors	Emotion (Text)
PS-Speech Interpretation	Speech Descriptors	Emotion (Speech)
PS-Face Interpretation	Face Descriptors	Emotion (Face)
Multimodal Emotion Fusion	Emotion (Text) Emotion (Speech) Emotion (Face)	Fused Emotion
Entity Dialogue Processing	1. Text Descriptors 2. Based on Input Selector 2.1. Refined Text 2.2. Input Text 3. Input Emotion	1. Machine Text 2. Machine Emotion
Text-to-Speech	1. Machine Text 2. Machine Emotion	Output Speech.

Video Lip Animation	1. Machine Emotion 2. Machine Speech	Output Visual

6.1.4.6 AIW, AIMs, and JSON Metadata

Table 4 – AIMs and JSON Metadata

AIW	AIMs	Name	JSON
MMC-CWE		Conversation With Emotion	X
	MMC-ASR	Automatic Speech Recognition	X
	MMC-EDP	Entity Dialogue Processing	X
	MMC-ESD	Entity Speech Description	X
	MMC-MEF	Multimodal Emotion Fusion	X
	MMC-NLU	Natural Language Understanding	X
	MMC-PFI	PS-Face Interpretation	X
	MMC-PTI	PS-Text Interpretation	X
	MMC-SPE	Speech Personal Status Extraction	X
	MMC-TTS	Text-to-Speech	X
	MMC-VLA	Video Lip Animation	X
	PAF-EFD	Entity Face Description	X
	PAF-FPE	Face Personal Status Extraction	X
	PAF-PSI	PS-Speech Interpretation	X

6.1.4.7 Conformance Testing

Table 5 provides the Conformance Testing Method for MMC-CWE 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 MMC-CWE AIM

Receives	Input Selector	Shall validate against Selector Schema.
	Text Object	Shall validate against Text Object Schema. Speech Data shall conform with Text Qualifier.
	Speech Object	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.
	Face Visual Object	Shall validate against Visual Object Schema. Face Data shall conform with Visual Qualifier.
Produces	Text Object	Shall validate against Text Object Schema. Speech Data shall conform with Text Qualifier.
	Speech Object	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.
	Face Visual Object	Shall validate against Visual Object Schema. Face Data shall conform with Visual Qualifier.

Important note. MMC-CWE Conformance Testing Specification V1.0 does not provide methods and datasets to Test the Conformance of:

1. Input Speech Description and PS-Speech Interpretation, but only of the Speech Personal Status Extraction (MMC-SPE) Composite AIM for Emotion.
2. Input Face Description and PS-Face Interpretation, but only of the Face Personal Status Extraction Composite AIM for Emotion.

Table 5 gives the input/output data of the MMC-CWE AI Workflow.

Table 5 - I/O data of MMC-CWE

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.
Speech Object	.wav	All input Speech files to be drawn from Speech files.
Face Object	AVC	All input Video files to be drawn from Video 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.
Machine Video	AVC	All Video files produced shall conform with Video files.

6.1.5 Human-CAV Interaction

6.1.5.1 Functions

The Human-CAV interaction (HCI) Subsystem has the function to recognise the human owner or renter, respond to humans' commands and queries, converse with humans, manifests itself as a perceptible entity, exchange information with the Autonomous Motion Subsystem in response to humans' requests, and communicate with HCIs on board other CAVs.

6.1.5.2 Reference Model

Figure 1 represents the Human-CAV Interaction (HCI) Reference Model.
It is assumed that Natural Language Understanding produces a Refined Text that is either the refined Recognised Text or the direct Input Text, depending on which one is being used.
Meaning is always computed based on the available text - Refined or Input.

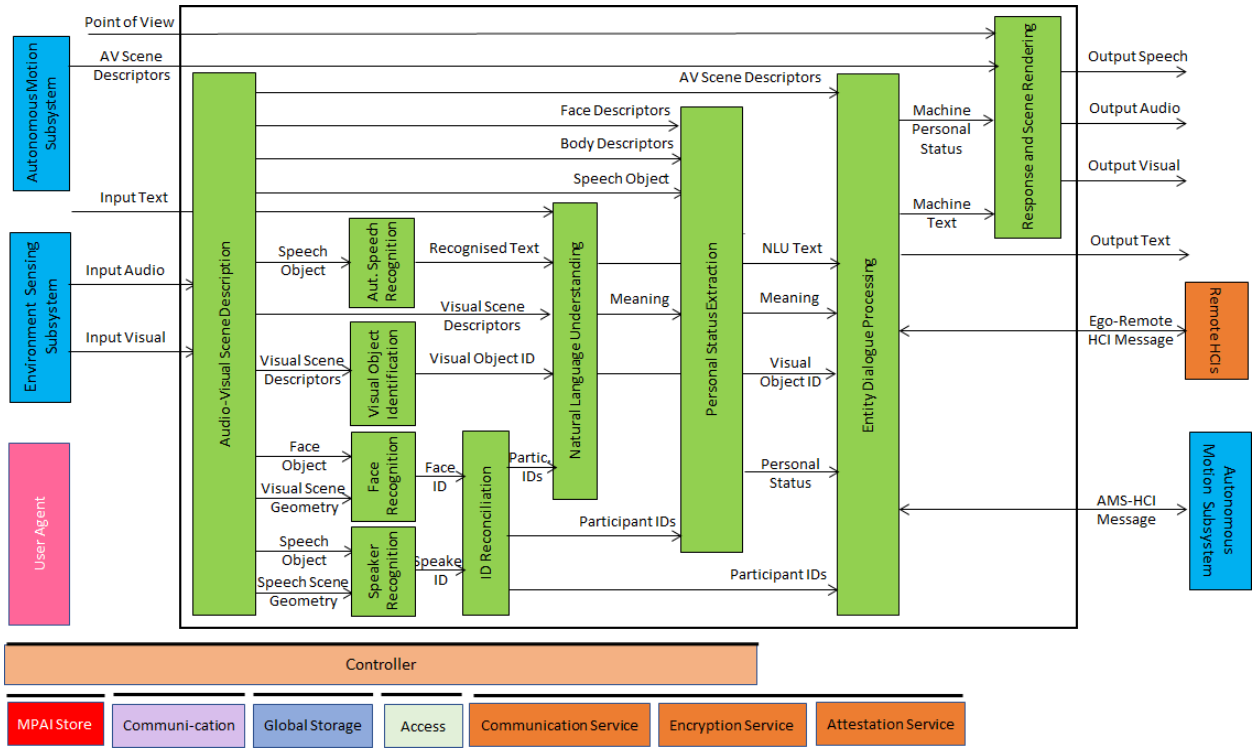

Figure 1 – Human-CAV Interaction Reference Model

The operation of the HCI subsystem is described by the following scenario where a group of humans approaches the CAV outside the CAV or is sitting inside the CAV:

1. Audio-Visual Scene Description (AVS) produces:
    1. Speech Scene Descriptors in the form of Speech Objects corresponding to each speaking human in the Environment (outside or inside the CAV)..
    2. Visual Scene Descriptors in the form of Descriptors of Faces and Bodies.
    3. All non-Speech Objects are removed from or signalled in the Audio Scene.
2. Automatic Speech Recognition (ASR) recognises the speech of each human and produces Recognised Text supporting multiple Speech Objects as input properly identified by the Spatial Attitude.
3. Visual Object Identification (VOI) produces Instance IDs of Visual Objects indicated by humans.
4. Natural Language Understanding (NLU)produces Refined Text and extracts Meaning from the Recognised Text of each Input Speech using the spatial information of Visual Object Identifiers.
5. Speaker Identity Recognition (SIR) and Face Identity Recognition (FIR) identifies the humans the HCI is interacting with. If FIR provides Face IDs corresponding to the Speaker IDs, Entity Dialogue Processing AIM can correctly associate the Speaker IDs (and the corresponding Text) with the Face IDs.
6. Personal Status Extraction (PSE) extracts the Personal Status of the humans.
7. Entity Dialogue Processing (EDP)
    1. Communicates with the Autonomous Motion Subsystem of the Ego CAV to request to:
        1. Move the CAV to a destination.
        2. Views the Full Environment Descriptors for the passengers' benefit.
        3. Be informed about CAV's situation.
        4. Receive relevant information for passengers.
    2. Communicates with the Autonomous Motion Subsystems of Remote CAVs.
    3. Produces the Machine Text and Machine Personal Status.

8. Personal Status Display (PSD) produces the Machine Portable Avatar conveying Machine Speech, Machine Personal Status, and any other information that may be relevant for the the Audio-Visual Rendering AIM .
9. Audio-Visual Scene Rendering (AVR) renders Audio, and Visual information using Machine Portable Avatar or the Autonomous Motion Subsystem's Full Environment Descriptors based on the Point of View provided by the human.
10. Entity Dialogue Processing (EDP)
    1. Requests the AMS subsystem to provide candidate Routes in response to a human requesting to be taken to a destination.
    2. Responses from AMS are processed by EDP and converted to multimodal messages understandable by the human.
    3. Eventually, the human accepts the Route or further elaborates on the EDP response.
    4. May receive messages from Ego AMS or Remote HCI that are processed and converted to multimodal messages understandable by the human.

The HCI interacts with the humans in the cabin in several ways:

1. By responding to commands/queries from one or more humans at the same time, e.g.:
    1. Commands to go to a waypoint, park at a place, etc.
    2. Commands with an effect in the cabin, e.g., turn off air conditioning, turn on the radio, call a person, open window or door, search for information etc.
2. By conversing with and responding to questions from one or more humans at the same time about travel-related issues (in-depth domain-specific conversation), e.g.:
    1. Humans request information, e.g., time to destination, route conditions, weather at destination, etc.
    2. CAV offers alternatives to humans, e.g., long but safe way, short but likely to have interruptions.
    3. Humans ask questions about objects in the cabin.
3. By following the conversation on travel matters held by humans in the cabin if
    1. The passengers allow the HCI to do so, and
    2. The processing is carried out inside the CAV.

6.1.5.3 I/O Data

Table 1 gives the input/output data of Human-CAV Interaction. I/O Data to/from Remote HCI and Ego AMS are not part of this Technical Specification.

Table 1 - I/O data of Human-CAV Interaction

Input data	From	Comment
Point of View	Passenger	Passenger's Point of View looking at environment.
Audio-Visual Scene Descriptors	AMS Subsystem	Audio-Visual representation of the environment.
Input Audio	Environment, Passenger Cabin	User authentication, command/interaction with HCI, etc. and environment Audio.
Input Text	User	Text complementing/replacing User input
Input Visual	Environment, Passenger Cabin	Environment perception, User authentication, command/interaction with HCI, etc. and environment Visual.
AMS-HCI Message	AMS Subsystem	AMS response to HCI request.

Ego-Remote HCI Message	Remote HCI	Remote HCI to Ego HCI.
Output data	**To**	**Comment**
Output Text	Cabin Passengers	HCI's avatar Text.
Output Speech	Cabin Passengers	HCT's avatar Speech.
Output Audio	Cabin Passengers	HCI's avatar or FED Audio.
Output Visual	Cabin Passengers	HCI's avatar or FED Visual.
AMS-HCI Message	AMS Subsystem	HCI request to AMS, e.g., Route or Point of View.
Ego-Remote HCI Message	Remote HCI	Ego HCI to Remote HCI.

6.1.5.4 Functions of AI Modules

Table 2 gives the functions of all Human-CAV Interaction AIMs.

Table 2 – Functions of Human-CAV Interaction's AI Modules

AIM	Function
Audio-Visual Scene Description	1. Receives Audio and Visual Objects from the appropriate Devices. 2. Produces Audio-Visual Scene Descriptors.
Automatic Speech Recognition	1. Receives Speech Objects. 2. Produces Recognised Text.
Visual Object Identification	1. Receives Visual Scenes Descriptors. 2. Provides Instance ID of indicated Visual Object.
Natural Language Understanding	1. Receives Recognised Text. 2. Uses context information (e.g., Instance ID of object). 3. Produces Natural Language Understanding Text (using Refined or Input) and Meaning.
Speaker Identity Recognition	1. Receives Speech Object of a human and Speech Scene Geometry. 2. Produces Speaker ID.
Personal Status Extraction	1. Receives Speech Object, Meaning, Face Descriptors and Body Descriptors of a human with a Participant ID. 2. Produces the human's Personal Status.
Face Identity Recognition	1. Receives Face Object of a human and Visual Scene Geometry. 2. Produces Face ID.
Entity Dialogue Processing	1. Receives Speaker ID, Face ID, AV Scene Descriptors, Meaning, Natural Language Understanding Text , Visual Object ID, and Personal Status. Moreover it receives AMS-HCI Messages and Ego-Remote HCI Messages. 2. Produces Machine (HCI) Text Object and Personal Status. Moreover it produces AMS-HCI Messages and Ego-Remote HCI Messages.

Personal Status Display	1. Receives Machine Text Object and Machine Personal Status. 2. Produces Machine's Portable Avatar.
Audio-Visual Scene Rendering	1. Receives AV Scene Descriptors, Portable Avatar, and Point of View. 2. Produces Output Speech, Output Audio, and Output Visual.

6.1.5.5 I/O Data of AI Modules

Table 3 gives the AI Modules of the Human-CAV Interaction depicted in Figure 3.

Table 3 – AI Modules of Human-CAV Interaction AIW

AIM	Input	Output
Audio-Visual Scene Description	- Input Audio - Input Visual	- AV Scene Descriptors
Automatic Speech Recognition	- Speech Object	- Recognised Text
Visual Object Identification	- AV Scene Descriptors - Visual Objects	- Visual Object Instance ID
Natural Language Understanding	- Recognised Text - AV Scene Descriptors - Visual Object Instance ID - Input Text	- Natural Language Understanding Text - Meaning
Speaker Identity Recognition	- Speech Object - Speech Scene Geometry	- Speaker ID
Personal Status Extraction	- Meaning - Input Speech - Face Descriptors - Body Descriptors	- Personal Status
Face Identity Recognition	- Face Object - Visual Scene Geometry	- Face ID
Entity Dialogue Processing	- Ego-Remote HCI Message - AMS-HCI Message - Speaker ID - Meaning - Natural Language Understanding Text - Visual Object Instance ID - Personal Status - Face ID	- Ego-Remote HCI Message - AMS-HCI Message - Machine Text - Machine Personal Status
Personal Status Display	- Machine Personal Status - Machine Text	- Machine Portable Avatar
Audio-Visual Scene Rendering	- AV Scene Descriptors - Machine Portable Avatar - Point of View	- Output Text - Output Speech - Output Audio - Output Visual

6.1.5.6 AIW, AIMs and JSON Metadata

Table 4 provides the links to the AIW and AIM specifications and to the JSON syntaxes. AIMs/1 indicates that the column contains Composite AIMs and AIMs/2 indicates that the column contains Basic and Composite AIMs. AIMs/3 indicates the the column only contains Basic AIMs.

Table 4 – AIMs and JSON Metadata

AIW	AIMs/1	AIMs/2	AIMs/3	Name	JSON
MMC-HCI				Human-CAV Interaction	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-ADM	Audio Descriptors Multiplexing	X
		OSD-VSD		Visual Scene Description	X
	MMC-ASR			Automatic Speech Recognition	X
	OSD-AVA			Audio-Visual Alignment	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
	MMC-NLU			Natural Language Understanding	X
	MMC-SIR			Speaker Identity Recognition	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-EDP			Entity Dialogue Processing	X
	PAF-FIR			Face Identity Recognition	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
	PAF-AVR			Audio-Visual Scene Rendering	X

6.1.5.7 Conformance Testing

Table 5 provides the Conformance Testing Method for MMC-HCI 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 MMC-HCI AIM

Receives	Input Audio	Shall validate against Audio Object Schema. Audio Data shall conform with Audio Qualifier.
	Input Text	Shall validate against Text Object Schema. Speech Data shall conform with Text Qualifier.
	Input Visual	Shall validate against Visual Object Schema. Speech Data shall conform with Visual Qualifier.
	AMS-HCI Message	Shall validate against AMS-HCI Message Schema.
	Ego-Remote HCI Message	Shall validate against Ego-Remote HCI Message Schema.
Produces	Output Text	Shall validate against Text Object Schema. Text Data shall conform with Text Qualifier.
	Output Speech	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.
	Output Audio	Shall validate against Audio Object Schema. Audio Data shall conform with Audio Qualifier.
	Output Visual	Shall validate against Visual Object Schema. Visual Data shall conform with Visual Qualifier.
	AMS-HCI Message	Shall validate against AMS-HCI Message Schema.
	Ego-Remote HCI Message	Shall validate against Ego-Remote HCI Message Schema.

6.1.6 Multimodal Question Answering

6.1.6.1 Functions

In a Question Answering (QA) System, a machine provides answers to a user's question presented in natural language. Multimodal Question Answering improves current QA systems

that are only able to deal with text or speech inputs by offering the requesting human the ability to present both speech or text and images. For example, users might ask "Where can I buy this tool?" while showing the picture of the tool, even without showing their faces. In the Multimodal Question Answering (MMC-MQA) Use Case, a machine responds to a question expressed by a user in text or speech while showing an object. The machine's response may use text and synthetic speech.

6.1.6.2 Reference Model

Figure 1 gives the Multimodal Question Answering Reference Model including the input/output data, the AIMs, and the data exchanged between and among the AIMs.

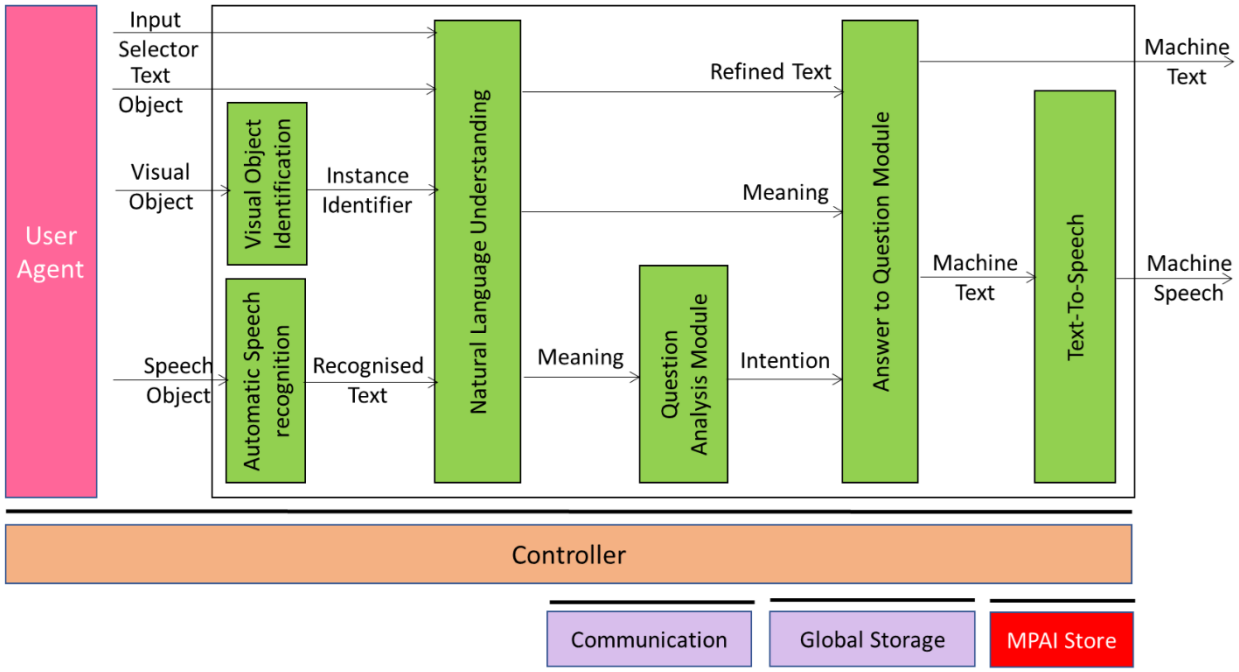

Figure 1 – Reference Model of Multimodal Question Answering

The operation of Multimodal Question Answering develops in the following way:

1. Input Selector is used to inform the machine whether the human employs Text or Speech to query the machine.
2. Depending on the value of Input Selector, Natural Language Understanding:
   - Extracts the Meaning of the question from and refines Recognised Text.
   - Extracts the Meaning of the question from Input Text.
3. Visual Object Identification identifies the Visual Object.
4. Question Analysis Module determines the Intention of the question.
5. Module for Question Answering uses Intention and Meaning to produce the answer as Machine Text.
6. Text-To-Speech produces Machine Speech from Machine Text.

6.1.6.3 I/O Data

The input and output data of the Multimodal Question Answering Use Case are:

Table 1 – I/O Data of Multimodal Question Answering

Input	Descriptions
Text Object	Text typed by the human as a replacement for Input Speech.

Input Selector	Data determining the use of Speech or Text.
Visual Object	Video of the human showing an object held in hand.
Speech Object	Speech of the human asking a question the Machine.
Output	**Descriptions**
Machine Text	The Text generated by Machine in response to human input.
Machine Speech	The Speech generated by Machine in response to human input.

6.1.6.4 Functions of AI Modules

Table 2 provides the functions of the Multimodal Question Answering Use Case.

Table 2 – Functions of AI Modules of Multimodal Question Answering

AIM	Function
Visual Object Identification	Identifies the Visual Object.
Automatic Speech Recognition	Recognises Speech.
Natural Language Understanding	Extracts Meaning and refines Text from Recognised Text.
Question Analysis Module	Extracts Intention from Text.
Answer to Question Module	Produces response of Machine to the query.
Text-to-Speech	Synthesises Speech from Text.

6.1.6.5 I/O Data of AI Modules

The AI Modules of Multimodal Question Answering are given in *Table 3*.

Table 3 – AI Modules of Multimodal Question Answering

AIM	Receives	Produces
Visual Object Identification	Visual Object	Instance Identifier
Automatic Speech Recognition	Speech Object	Recognised Text
Natural Language Understanding	Text Object or Recognised Text	Refined Text Meaning
Question Analysis Module	Meaning	Intention
Answer to Question Module	1. Input or Recognised Text 2. Intention 3. Meaning	Machine Text
Text-to-Speech	Machine Text	Machine Speech

6.1.6.6 AIW, AIMs, and JSON Metadata

Table 4 provides the links to the AIW and AIM specifications and to the JSON syntaxes. AIMs/1 indicates that the column contains Composite AIMs and AIMs/2 indicates that the column contains their Basic AIMs.

Table 4 – AIW, AIMs, and JSON Metadata

AIW	AIMs/1	AIMs/2	Name	JSON
MMC-MQA			Multimodal Question Answering	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
	MMC-ASR		Automatic Speech Recognition	X
	MMC-NLU		Natural Language Understanding	X
	MMC-QAM		Question Analysis Module	X
	MMC-AQM		Answer to Question Module	X
	MMC-TTS		Text-to-Speech	X

6.1.6.7 Conformance Testing

Table 5 provides the Conformance Testing Method for MMC-MQA 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 MMC-MQA AIM

Receives	Text Object	Shall validate against Text Object Schema. Speech Data shall conform with Text Qualifier.
	Input Selector	Shall validate against Selector Schema.
	Visual Object	Shall validate against Visual Object Schema. Speech Data shall conform with Visual Qualifier.
	Speech Object	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.
Produces	Machine Text	Shall validate against Text Object Schema. Speech Data shall conform with Text Qualifier.
	Machine Speech	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.

Table 6 provides an example of MMC-AQM AIM conformance testing.

Table 6 – An example MMC-MQA AIM conformance testing

Input Data	Data Type	Input Conformance Testing Data
Input Selector	Binary data	For Input Selector=0 and Input Selector=1
Text Object	Unicode	All input Text files to be drawn from Text Files.
Speech Object	.wav	All input Speech files to be drawn from Audio Files.
Input Image	JPEG	All input Image files to be drawn from Images.

Output Data	Data Type	Input Conformance Testing Data
Machine Text	Unicode	All Text files produced shall conform with Text files.
Machine Speech	.wav	All Speech files produced shall conform with Speech files.

6.1.7 Text and Speech Translation

6.1.7.1 Functions

The goal of the Text and Speech Translation (MMC-UST) Use Case is to translate speech segments expressed in a source language into a target language or to produce the textual version of the translated speech. If the desired output is speech, the user can specify whether their speech features (voice colour, emotional charge, etc.) should be preserved in the translated speech.
The flow of control is from Input Speech or Input Text to Translated Text, and then to Output Speech and Output Text. Depending on the value of Input Selector:

1. Input Text in Language A is translated into Translated Text in Language B and pronounced as Speech in Language B.
2. The Speech features (voice colour, emotional charge, etc.) in Language A are preserved in Language B.

6.1.7.2 Reference Model

Figure 1 depicts the input/output data, the AIMs, and the data exchanged between AIMs of the Text and Speech Translation AIW.

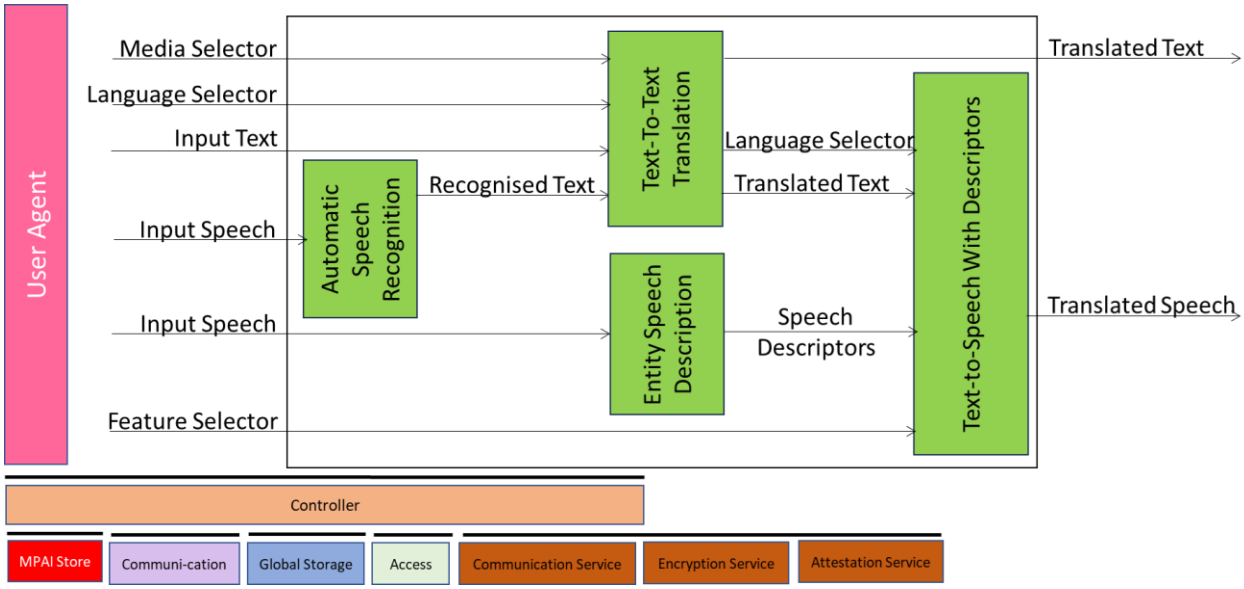

Figure 1 – Reference Model of Text and Speech Translation (MMC-TST) AIW

In previous MPAI-MMC versions, this AIW was called Unidirectional Speech Translation (MMC-UST). The same previous versions included two variations of the Text and Speech Translation (MMC-TST): Bidirectional and One-to-Many. They are reported here to show that they are based on the same MMC-TST AI Workflow.

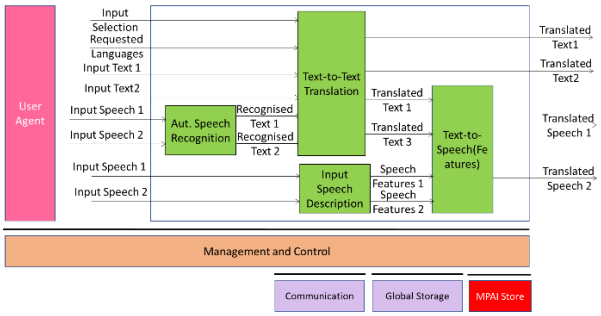

Figure 2 - Bidirectional Speech Translation (MMC-BST) V2.1

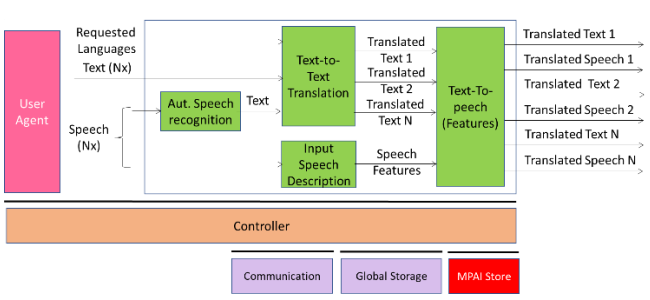

Figure 2 - One-to-Many Speech Translation (MMC-MST) V2.1

6.1.7.3 I/O Data

The input and output data of the Text and Speech Translation AI Workflow are:

Table 1 – I/O Data of Text and Speech Translation

Input	Descriptions
Media Selector	Determines whether the input will be in Text or Speech
Language Selector	Determines the input and output language
Feature Selector	Determines whether the Speakers vocal features should be added to synthetic speech.
Input Speech	Speech produced in Language A by a human desiring translation into language B.
Input Text	Alternative textual source information to be translated into and pronounced in language B depending on the value of Input Selector.
Media Selector	Determines whether: the Input Speech features are preserved in the Output Speech.
Output	**Descriptions**
Translated Speech	Input Speech translated into language B preserving the Input Speech features in the Output Speech, depending on the value of Input Selector.
Translated Text	Text of Input Speech or Input Text translated into language B, depending on the value of Input Selector.

6.1.7.4 Functions of AI Modules

Table 2 gives the functions of Text and Speech Translation AIMs.

Table 2 – Functions of Text and Speech Translation AI Modules

AIM	Functions
Automatic Speech Recognition	Recognises Speech
Text-to-Text Translation	Translates Recognised Text
Entity Speech Description	Extracts Speech Features
Text-to-Speech With Descriptors	Synthesises Translated Text adding Speech Features

6.1.7.5 I/O Data of AI Modules of Text and Speech Translation

The AI Modules of Text and Speech Translation are given in *Table 3*.

Table 3 – AI Modules of Text and Speech Translation

AIM	Receives	Produces
Automatic Speech Recognition	Input Speech	Recognised Text
Text-to-Text Translation	1. Input Text 2. Recognised Text	Translated Text
Entity Speech Description	Input Speech	Speaker's Speech Descriptors
Text-to-Speech With Descriptors	1. Translated Text 2. Speech Descriptors	Produces Output Speech.

6.1.7.6 AIW, AIMs, and JSON Metadata

Table 4 - AIMs and JSON Metadata

AIW	AIMs	Name	JSON
MMC-TST		Text and Speech Translation	X
	MMC-ASR	Automatic Speech Recognition	X
	MMC-TTT	Text-to-Text Translation	X
	MMC-ESD	Entity Speech Description	X
	MMC-DTS	Text-to-Speech With Descriptors	X

6.1.7.7 Conformance Testing

Table 5 provides the Conformance Testing Method for MMC-TSTAIM.
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 MMC-TST AIM

Receives	Media Selector	Shall validate against Selector Schema.
	Language Selector	Shall validate against Selector Schema.
	Feature Selector	Shall validate against Selector Schema.
	Input Speech	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.
	Input Text	Shall validate against Text Object Schema. Speech Data shall conform with Text Qualifier.
	Media Selector	Shall validate against Selector Schema.
Produces	Translated Speech	Shall validate against Speech Object Schema. Speech Data shall conform with Speech Qualifier.
	Translated Text	Shall validate against Text Object Schema. Speech Data shall conform with Text Qualifier.

Table 6 provides an example of MMC-TST AIM conformance testing.

Table 6 – An example MMC-AQM TST conformance testing

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.

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.

6.1.8 Virtual Meeting Secretary

6.1.8.1 Functions

In Virtual Meeting applications, such as in the Avatar Based Videoconference, i.e., a video-conference where avatars participate realistically impersonating the human participants, the Virtual Secretary:

1. Listens to the Speech of each avatar.
2. Monitors their Personal Status.
3. Drafts a Summary using the avatars' Personal Status and Text obtained from Automatic Speech Recognition or directly via Text input in the meeting's common language.
4. The Summary can be made available to participants in two different ways:
    - Transferred to an external application so that participants can edit the Summary.
    - Displayed to avatars where:
        - Avatars make Speech or Text comments (e.g., offline via chat).
        - The Virtual Secretary edits the Summary by understanding Text, Speech, and the avatars' Personal Statuses.

6.1.8.2 Reference Model

Figure 1 specifies the Reference Model of the Virtual Secretary AIW. It is assumed that Meaning represents both meaning of Input Text and meaning of Refined Text.

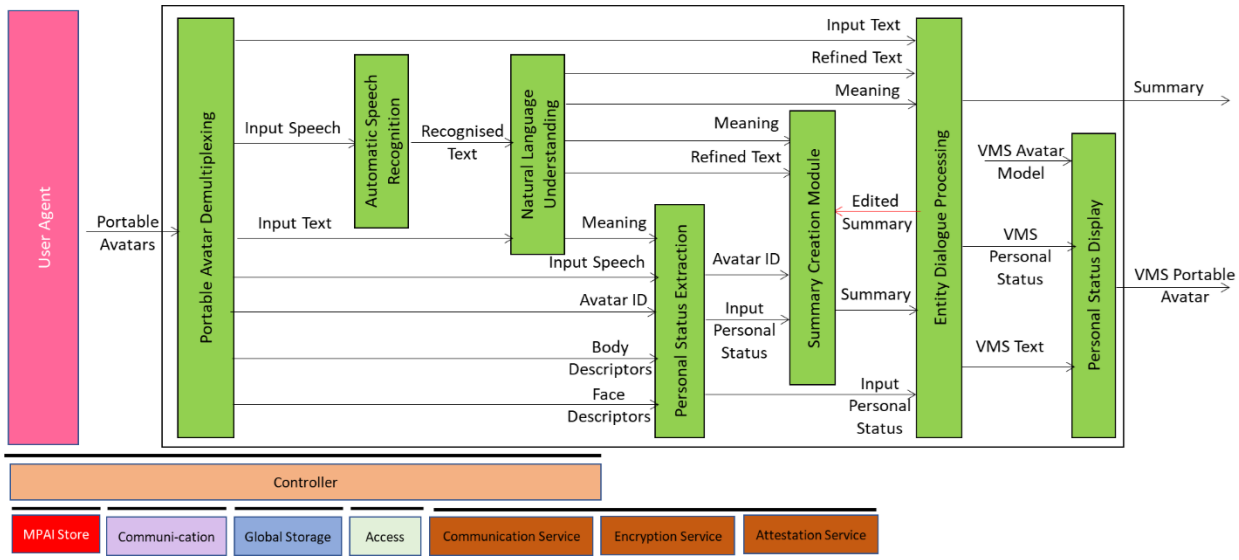

Figure 1 – Reference Model of the Virtual Meeting Secretary (MMC-VMS) AIW

The Virtual Secretary processes one avatar at a time according to the following workflow:

1. Portable Avatar De-multiplexing produces Input Text, Input Speech, Avatar ID, Body Descriptors, and Face Descriptors.
2. Automatic Speech Recognition extracts Text from Avatar Speech.
3. Natural Language Understanding:
    - Receives Recognised Text.
    - Produces Refined Text (of Recognised Text) and Meaning.
4. Personal Status Extraction:
    - Receives Meaning, Speech, and Body and Face Descriptors.
    - Produces the Personal Status of the avatar it is interacting with.
5. Summary Creation Module:
    - Receives Refined Text, Personal Status, and Meaning
    - Produces Summary using Personal Status and Text in the meeting's common language.
    - Receives Edited Summary from Entity Dialogue Processing.
6. Entity Dialogue Processing:
    - Sends Summary to external application.
    - Sends Edited Summary produced from Refined Text (from Speech), Avatar's Text (from chat), Meaning, and Summary back to Summary Creation Module.
    - Produces VMS Text and VMS Personal Status.
7. Personal Status Display produces VMS Portable Avatar containing VMS Avatar Model, VMS Text, VMS Speech, and VMS Avatar Descriptors.

6.1.8.3 I/O Data

Table 1 gives the input/output data of Virtual Meeting Secretary.

Table 1 – I/O data of Virtual Meeting Secretary

Input data	From	Description
Portable Avatar	Server	Participants' Portable Avatars as re-multiplexed by Server
Output data	**To**	**Descriptions**
VMS Portable Avatar	Server	VMS Portable Avatar to Server

Summary	Server	Summary of avatars' interventions

6.1.8.4 Functions of AI Modules

Table 2 gives the functions of Virtual Meeting Secretary AIMs.

Table 2 – Functions of Virtual Meeting Secretary AI Modules

AIM	Functions
Portable Avatar Demultiplexing	1. Receives Portable Avatar. 2. Provides the Data required by Virtual Secretary's AIMs.
Automatic Speech Recognition	1. Receives Speech. 2. Provides Recognised Text.
Natural Language Understanding	1. Refines Recognised Text. 2. Extracts Meaning.
Personal Status Extraction	1. Receives Meaning, Input Speech, Body Descriptors, Face Descriptors. 2. Extracts Personal Status.
Summary Creation Module	1. Receives Meaning, Refined Text, Avatar ID, Input Personal Status of Avatar ID, and Edited Summary (from Entity Dialogue Processing.) 2. Produces and refines Summary using Edited Summary.
Entity Dialogue Processing	1. Receives Input Text, Refined Text, Meaning, Summary, Input Personal Status. 2. Produces Text, Virtual Secretary Personal Status, and Edited Summary.
Personal Status Display	1. Receives Virtual Secretary's Avatar Model, Personal Status, and Text. 2. Shows Virtual Secretary as Virtual Secretary Portable Avatar.

6.1.8.5 I/O Data of AI Modules

Table 3 gives the AI Modules of the Virtual Secretary depicted in Figure 4.

Table 3 – AI Modules of Virtual Secretary

AIM	Receives	Produces
Portable Avatar Demultiplexing	Portable Avatar	1. Input Text 2. Input Speech 3. Avatar ID 4. Body Descriptors Object 5. Face Descriptors Object
Automatic Speech Recognition	Speech	Recognised Text
Natural Language Understanding	Recognised Text	1. Refined Text 2. Meaning
Personal Status Extraction	1. Meaning 2. Speech 3. Face Descriptors Object 4. Body Descriptors Object	Personal Status

Summary Creation Module	1. Meaning 2. Refined Text 3. Edited Summary	Summary
Entity Dialogue Processing	1. Refined Text 2. Portable Avatar 3. Meaning 4. Summary	1. VMS Portable Avatar 2. VMS Text 3. Edited Summary
Personal Status Display	1. VMS Text 2. VMS Avatar Model 3. VMS Personal Status	VMS Portable Avatar

6.1.8.6 AIW, AIMs, and JSON Metadata

Table 4 – AIMs and JSON Metadata

Note: AIM1/s are Composite AIMs, AIM/2s are Basic AIMs.

AIW	AIMs/1	AIMs/2	Name	JSON
MMC-VMS			Virtual Meeting Secretary	X
	PAF-PDX		Portable Avatar Multiplexing	X
	MMC-ASR		Automatic Speech Recognition	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-SCM		Summary Creation Module	X
	MMC-EDP		Entity Dialogue Processing	X
	PAF-PSD		Personal Status Display	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

6.1.8.7 Conformance Testing

Table 5 provides the Conformance Testing Method for MMC-VMS 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 MMC-VMS AIM

Receives	Portable Avatars	Shall validate against Portable Avatar Schema. Speech Data shall conform with Summary Qualifier.
Produces	VMS Portable Avatar	Shall validate against Portable Avatar Schema. Speech Data shall conform with Summary Qualifier.
	Summary	Shall validate against Summary Schema. Speech Data shall conform with Summary Qualifier.

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

Note that at this stage the MPAI-MMC AIWs implement only a part of the AIMs.

6.3 Conformance Testing

An implementation of an AI Workflow conforms with MPAI-MMC if it accepts as input and produces as output Data and/or Data Objects (the combination of Data of a Data Type and its Qualifier) conforming with those specified by MPAI-MMC.
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, and the Performance Assessment Specification should provide methods to measure how well an AIW performs its function, using a metric that depends on the nature of the function, such as:

1. *Quality*: the Performance of an Answer to Question Module AIW can measure how well the AIW answers a question related to an image.
2. *Bias*: Performance of an Answer to Question Module AIW can measure the quality of responses in dependence of the type of images.

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.

The current MPAI-MMC V2.3 Standard does not provide AIW Performance Assessment methods.

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: Multimodal Conversation (MPAI-MMC) V2.4***. All previously specified MPAI-MMC AI-Modules are superseded by those specified by V2.4 but may be used by explicitly signaling their version. Bold characters are used to indicate that an AIM is Composite.

Table 1 - Specifications and JSON syntax of AIMs used by MPAI-MMC V2.4

AIMs	Name	JSON	AIMs	Name	JSON
MMC-AQM	Answer to Question Module	X	MMC-PTI	PS-Text Interpretation	X
MMC-ASR	Automatic Speech Recognition	X	MMC-QAM	Question Analysis Module	X
MMC-AUS	Audio Segmentation	X	MMC-SCM	Summary Creation Module	X
MMC-EDP	Entity Dialogue Processing	X	MMC-SIR	Speaker Identity Recognition	X
MMC-ESD	Entity Speech Description	X	**MMC-SPE**	**Speech Personal Status Extraction**	**X**
MMC-ETD	Entity Text Description	X	**MMC-STD**	**Speech Translation with Descriptors**	**X**
MMC-MEF	Multimodal Emotion Fusion	X	MMC-TSD	Text-to-Speech with Descriptors	X
MMC-NLU	Natural Language Understanding	X	**MMC-TST**	**Text and Speech Translation**	**X**
MMC-PDX	Personal Status Demultiplexing	X	**MMC-TIQ**	**Text and Image Query**	**X**
MMC-PMX	Personal Status Multiplexing	X	MMC-TTS	Text-To-Speech	X
MMC-PSE	**Personal Status Extraction**	**X**	MMC-TTT	Text-to-Text Translation	X
MMC-PSI	PS-Speech Interpretation	X	MMC-VLA	Video Lip Animation	X

7.1.1 Answer to Question Module

7.1.1.1 Functions

The Answer to Question Module (MMC-AQM) AIM provides a text in response to an input text accompanied by the Meaning and Intention of the input text:

Receives	Text Object	Text of query
	Meaning	Meaning of query Text
	Intention	Intention of query Text
Produces	Text Object	of AIM's answer

7.1.1.2 Reference Model

Figure 1 specifies the Reference Module of the Answer To Question Module (MMC-AQM) AIM.

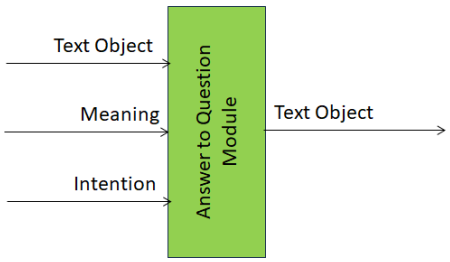

Figure 1 – The Answer To Question Module (MMC-AQM) AIM

7.1.1.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Answer To Question Module (MMC-AQM) AIM.

Table 1 – I/O Data of the Answer To Question Module (MMC-AQM) AIM

Input data	Description
Text Object	Input Text Object, e.g., from Natural Language Understanding
Meaning	Meaning of Text
Intention	Intention in Text
Output data	**Description**
Text Object	Text Object of Answer to Question

7.1.1.4 JSON Metadata

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

7.1.1.5 Conformance Testing

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

Input	Text Object	Shall validate against Text Object schema. Also, Text Data shall conform with Text Qualifier.
	Meaning	Shall validate against the Meaning schema.
	Intention	Shall validate against the Intention schema.
Output	Text Object	Shall validate against the Text Object schema. Also, Text Data shall conform with the Text Qualifier, e.g., the Language Format shall be that of input Language.

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**
Text Object	Unicode	All input Text files to be drawn from Question Text Files.
Meaning	JSON	All input Meaning files to be drawn from Meaning JSON Files for Questions.
Intention	JSON	All input Intention files to be drawn from Intention JSON Files.
Output Data	**Data Type**	**Output Conformance Testing Criteria**
Text Object	Unicode	All Text files produced shall conform with Text.

7.1.2 Automatic Speech Recognition

7.1.2.1 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.

7.1.2.2 Reference Model

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

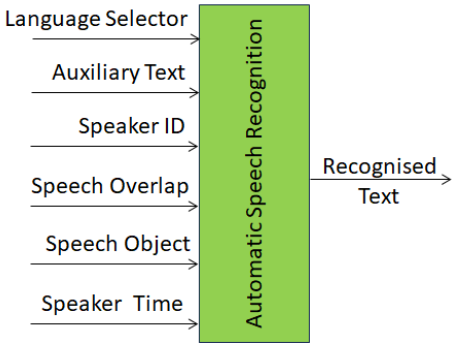

Figure 1 – The Automatic Speech Recognition (MMC-ASR) AIM

7.1.2.3 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.

7.1.2.4 JSON Metadata

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

7.1.2.5 Reference Software

7.1.2.5.1 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.

7.1.2.5.2 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

7.1.2.5.3 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).

7.1.2.5.4 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).

7.1.2.6 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.

7.1.2.7 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".

7.1.3 Audio Segmentation

7.1.3.1 Functions

The Audio Segmentation (MMC-AUS) AIM provides the speech segments that corresponds to different speaker in an input utterance, the corresponding start and end times of the speech segments, and the Speech Overlap data type of each speech segment:

Receives	*Audio Object*	Audio to be segmented including speech and audio.
Identifies	*Speech Time*	Time of audio segmentation.
Extracts	*Target Speech Objects*	Speech Object to be extracted.
Detects	*Speech Overlap*	Data Type about speech overlap.
Produces	*Speech Time*	Duration of speech segment.
	Speech Overlap	Data Type about speech overlap.
	Speech Objects	Each Speech Object includes a Speaker's Turn, i.e., one or more adjacent utterances from the same Speaker.

7.1.3.2 Reference Model

Figure 1 depicts the Reference Model of the Audio Segmentation (MMC-AUS) AIM.

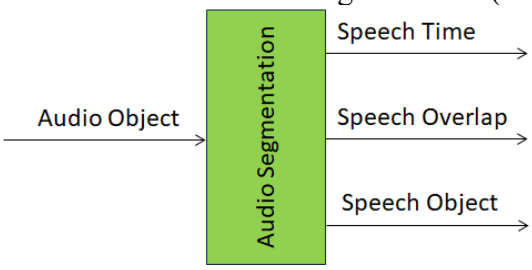

Figure 1 - Reference Model of Audio Segmentation (MMC-AUS) AIM

7.1.3.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio Segmentation (MMC-AUS) AIM.

Table 1 – I/O Data of the Audio Segmentation (MMC-AUS) AIM

Input	Description
Audio Object	Input Audio in a file.
Output	**Description**
Speaker Time	Time one or more Speakers start speaking.
Speech Overlap	Number of overlapping speakers.
Speech Object	Speech Object containing the utterance(s) of the Speaker(s).

7.1.3.4 JSON Metadata

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

7.1.3.5 Reference Software

7.1.3.5.1 Disclaimers

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

7.1.3.5.2 Guide to the code

MMC-AUS splits the input WAV file into speech segments - called speakers' turns - a belonging to one - still unidentified speaker. See see "start and end times of each speaker's turn, as well as the speaker labels" at https://www.aimodels.fyi/models/huggingFace/speaker-diarization-pyannote. A turn is defined as a sequence of one or more speech segments belonging to the same speaker. See https://dokumen.pub/speech-recognition-technology-and-applications-9798886971798.html.

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

The MMC-AUS 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 and https://huggingface.co/pyannote/segmentation
5. diar_conf.yaml: YML setting up a diarization pipeline. Copy it to $AI_FW_DIR/confs/mmc_aus

Library: https://github.com/pyannote/pyannote-audio

7.1.3.5.3 Acknowledgements

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

7.1.3.6 Conformance Testing

Table 2 provides the Conformance Testing Method for Audio Segmentation (MMC-AUS) 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 Audio Segmentation (MPAI-MMC) AIM

Input	Audio Object	Shall validate against Audio Object schema. Audio Data shall conform with Audio Qualifier.
Output	Speaker Time	Shall validate against Time schema.
	Speech Overlap	Shall validate against Speech Overlap schema.
	Speech Object	Shall validate against Speech Object schema.

7.1.3.7 Performance Assessment

Performance Assessment of an MMC-AUS AIM Implementation shall be performed using a dataset of Audio segments that has been annotated with Time and type of Data (single speech, overlapped speech, or overlapped audio and one or more speech segments).

The Performance Assessment Report of an MMC-AUS AIM Implementation shall include:

1. The Identifier of the MMC-AUS AIM.
2. The Identifier of the non-annotated dataset of audio segments (if available).
3. The Identifier of the annotated dataset of audio segments (if available).
4. The number of Audio segments in the data base (the number of segments in #2. and #3. shall be the same).
5. The Arithmetic Mean of
    1. The differences between Times of Speech Data computed by the MMC-AUS AIM and the annotated Times.
    2. Errors of more than 0.2 seconds in the identification of Speech Data Times.

7.1.4 Entity Dialogue Processing

7.1.4.1 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.

7.1.4.2 Reference Model

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

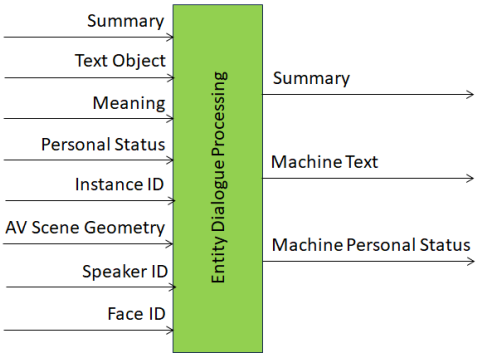

Figure 1 – Entity Dialogue Processing (MMC-EDP) AIM Reference Model

7.1.4.3 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.

7.1.4.4 JSON Metadata

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

7.1.4.5 Profiles

Profiles of Entity Dialogue Processing are specified.

7.1.4.6 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.

7.1.5 Entity Speech Description

7.1.5.1 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.

7.1.5.2 Reference Model

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

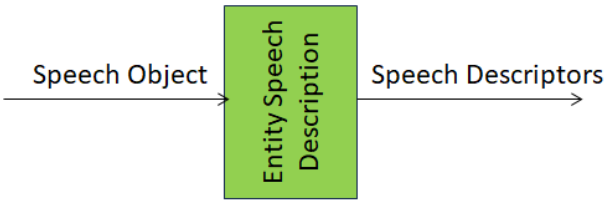

Figure 1 Entity Speech Description (MMC-ESD) AIM Reference Model

7.1.5.3 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

7.1.5.4 JSON Metadata

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

7.1.5.5 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.

7.1.6 Entity Text Description

7.1.6.1 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.

7.1.6.2 Reference Model

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

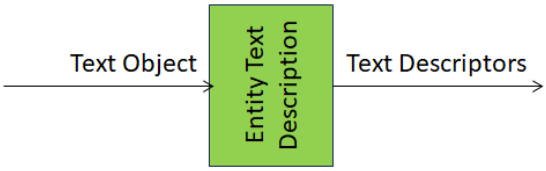

Figure 1 Entity Text Description (MMC-ETD) AIM Reference Model

7.1.6.3 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.

7.1.6.4 JSON Metadata

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

7.1.6.5 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.

7.1.7 Multimodal Emotion Fusion

7.1.7.1 Functions

The Multimodal Emotion Fusion (MMC-MEF) AIM receives the emotion of text, the emotion of speech, and the emotion of face and produces the total emotion:

Receives	Entity's Emotion (Text)	Emotion in Text
	Entity's Emotion (Speech)	Emotion in Speech
	Entity's Emotion (Face)	Emotion in Face
Produces	Entity's Emotion	Emotion

7.1.7.2 Reference Model

Figure 1 depicts the Reference Model of the Multimodal Emotion Fusion (MMC-MEF) AIM.

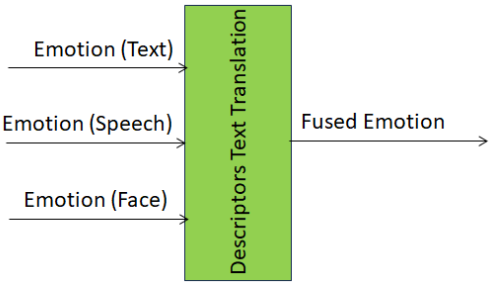

Figure 1 – The Multimodal Emotion Fusion (MMC-MEF) AIM Reference Model

7.1.7.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Multimodal Emotion Fusion (MMC-MEF) AIM.

Table 1 – I/O Data of the Multimodal Emotion Fusion (MMC-MEF) AIM

Input data	**Description**
Emotion (Text)	Emotion in Text
Emotion (Speech)	Emotion in Speech
Emotion (Face)	Emotion in Face
Output data	**Description**
Input Emotion	The estimated emotion that fuses all inputs

7.1.7.4 JSON Metadata

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

7.1.7.5 Conformance Testing

Table 2 provides the Conformance Testing Method for Multimodal Emotion Fusion (MMC-MEF) 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-MEF AIM

Input	Emotion (Text)	Shall validate against Emotion schema.
	Emotion (Speech)	Shall validate against Emotion schema.
	Emotion (Face)	Shall validate against Emotion schema.
Output	Input Entity's Emotion	Shall validate against Emotion schema.

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

Table 3 – An example MMC-MEF AIM conformance testing

Input Data	**Data Type**	**Input Conformance Testing Data**
Emotion (Text)	JSON	All JSON Emotion (Text) JSON files to be drawn from Emotion JSON Files
Emotion (Speech)	JSON	All Emotion (Speech) JSON files to be drawn from Emotion JSON Files
Emotion (Visual)	JSON	All Emotion (Face) JSON files to be drawn from Emotion JSON Files
Output Data	**Data Type**	**Output Conformance Testing Criteria**
Input Emotion	JSON	All Emotion JSON File 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.

7.1.8 Natural Language Understanding

7.1.8.1 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).

7.1.8.2 Reference Model

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

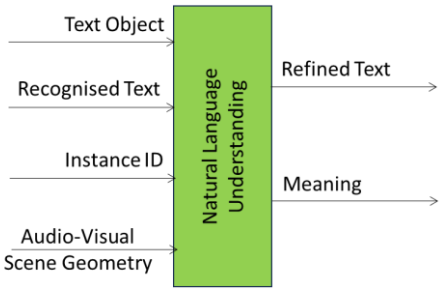

Figure 1 – The Natural Language Understanding (MMC-NLU) AIM Reference Model

7.1.8.3 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.

7.1.8.4 JSON Metadata

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

7.1.8.5 Profiles

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

7.1.8.6 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.

7.1.9 Personal Status Demultiplexing

7.1.9.1 Functions

The Personal Status Demultiplexing (MMC-PDX) AIM receives an instance of the Personal Status data type and demultiplexes its components elements: Text Personal Status, Speech Personal Status, Face Personal Status, and Gesture Personal Status:

Receives	*Personal Status*	An instance of the Personal Status Data Type.
Produces	*PS-Text*	Personal Status of Text Object.
	PS-Speech	Personal Status of Speech Object.
	PS-Face	Personal Status of Face Object.
	PS-Gesture	Personal Status of Gesture conveyed by Body Object.

7.1.9.2 Reference Architecture

Figure 1 depicts the Reference Architecture of the Personal Status Demultiplexing (MMC-PDX) AIM.

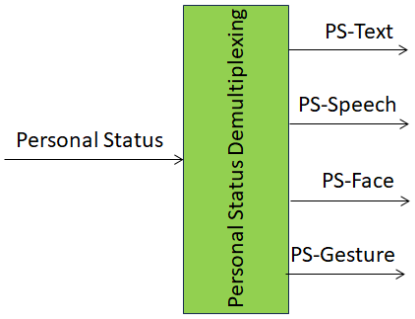

Figure 1 – The Personal Status Demultiplexing (MMC-PDX) AIM

7.1.9.3 I/O Data

Table 1 specifies the Input and Output Data of the Personal Status Multiplexing AIM.

Table 1 – I/O Data of the Personal Status Demultiplexing (MMC-PDX)

Input	Description
Personal Status	Input Personal Status.
Output	**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.

7.1.9.4 JSON Metadata

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

7.1.10 Personal Status Multiplexing

7.1.10.1 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

7.1.10.2 Reference Model

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

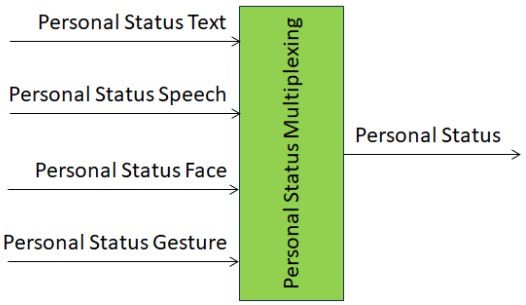

Figure 1 – The Personal Status Multiplexing (MMC-PSM) AIM Reference Model

7.1.10.3 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.

7.1.10.4 JSON Metadata

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

7.1.10.5 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.

7.1.11 Personal Status Extraction

7.1.11.1 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	

7.1.11.2 Reference Model

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

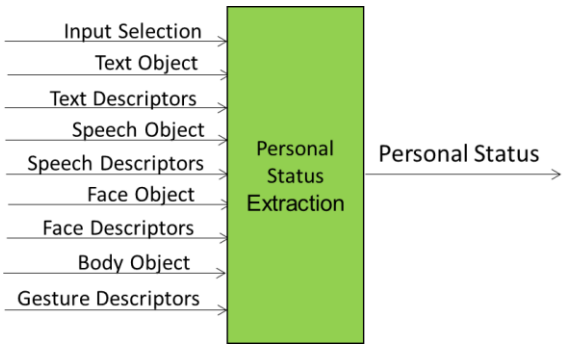

Figure 1 – The Personal Status Extraction Composite (MMC-PSE) AIM Reference Model

7.1.11.3 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

7.1.11.4 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

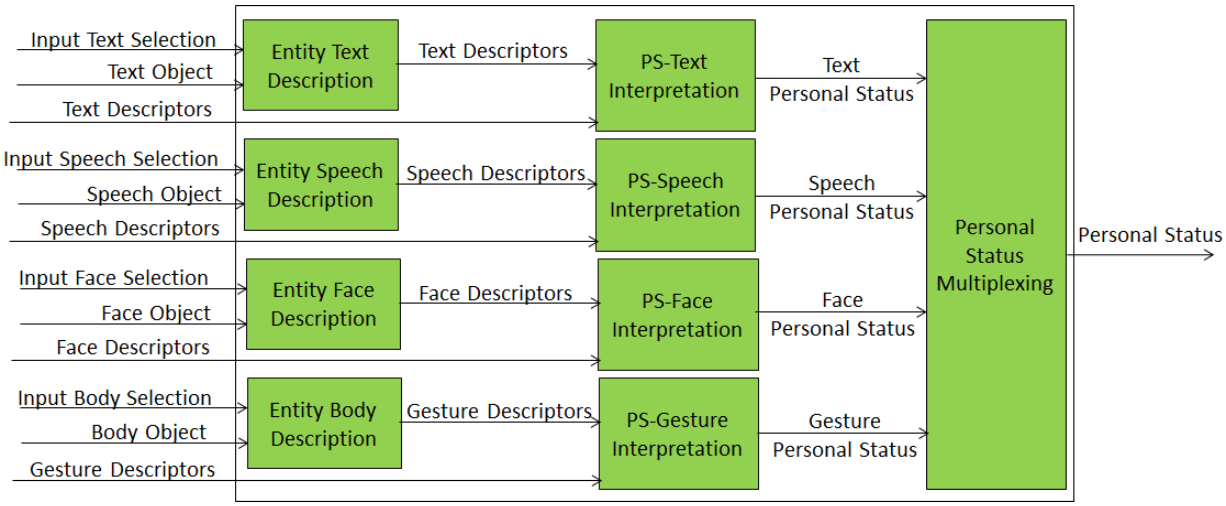

Figure 2 - The version of Personal Status Extraction AIM with the highest level of composition.

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

7.1.11.5 JSON Metadata

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

7.1.11.6 Profiles

The Profiles of Personal Status Extraction are specified.

7.1.11.7 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.

7.1.12 PS-Speech Interpretation

7.1.12.1 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.

7.1.12.2 Reference Model

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

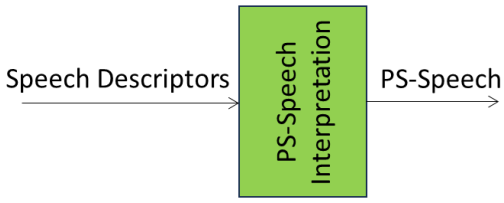

Figure 1 The PS-Speech Interpretation (MMC-PSI) AIM Reference Model

7.1.12.3 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

7.1.12.4 JSON Metadata

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

7.1.12.5 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

7.1.13 PS-Text Interpretation

7.1.13.1 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.

7.1.13.2 Reference Model

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

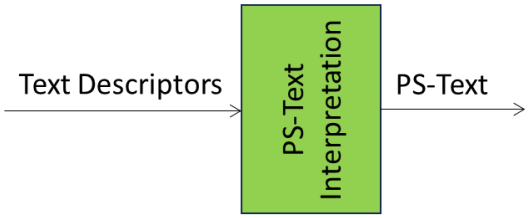

Figure 1 – The PS-Text Interpretation (MMC-PRI) AIM Reference Model

7.1.13.3 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

7.1.13.4 JSON Metadata

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

7.1.13.5 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

7.1.14 Question Analysis Module

7.1.14.1 Functions

The Question Analysis Module (MMC-QAM) AIM receives the Meaning of an input text and produces the Intention of the input text:

Receives	*Meaning*	Of Question
Produces	*Intention*	Of Question

7.1.14.2 Reference Model

Figure 1 depicts the Reference Module of the Question Analysis Module (MMC-QAM) AIM.

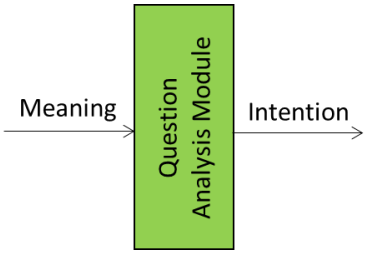

Figure 1 – The Question Analysis Module AIM Reference Module (MMC-QAM)

7.1.14.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Question Analysis Module (MMC-QAM) AIM.

Table 1 – I/O Data of the Question Analysis Module (MMC-QAM) AIM

Input data	Description
Meaning	Result of analysis Question's Text.
Output data	**Description**
Intention	Intention in Question.

7.1.14.4 JSON Metadata

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

7.1.14.5 Conformance Testing

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

Input	Meaning	Shall validate against Meaning schema.
Output	Intention	Shall validate against Intention schema.

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

Table 3 – An example MMC-QAM AIM conformance testing

Input Data	**Data Type**	**Input Conformance Testing Data**
Meaning	.wav	All input Meaning files to be drawn from Meaning JSON Files.
Output Data	**Data Type**	**Output Conformance Testing Criteria**
Intention	Unicode	All Intention JSON files shall validate against the Intention JSON Schema.

qtopic, qfocus, qLAT, qSAT, and qdo-main must be present in the output JSON file of Intention. The value of any of the five attributes may be null.

7.1.15 Summary Creation Module

7.1.15.1 Functions

The Summary Creation Module (MMC-SCM) AIM receives an input text and produces a Summary of that text. The MMC-SCM AIM may also receive as input the identifier of the Entity producing the input text, the space-time information and the Personal Status of that Entity:

Receives	Entity ID	ID of Entity of which a report is made.
	Text Object	Text Object whose Data is reported Text.
	Space-Time	Entity's space-time information.
	Personal Status	Entity's Personal Status
	Summary	Summary produced
Produces	Summary	Edited summary sent back to Entity making the summary.

7.1.15.2 Reference Model

The Reference Model of the Summary Creation Module (MMC-SCM)is depicted in Figure 1.

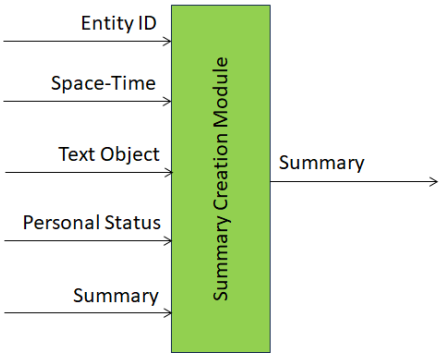

Figure 1 – Summary Creation Module (MMC-SCM)

7.1.15.3 Input/Output Data

Table 1 specifies the Input and Output Data of the .

Table 1 – I/O Data of the Summary Creation Module (MMC-SCM) AIM

Input	Description
Entity Identifier	ID of reported Entity.
Text Object	Text of reported Entity.
Personal Status	Avatar's Personal Status.
Edited Summary	The Summary revised by the Dialogue Processing.
Space-Time	The Entity's Space and Time information.
Outputs	**Description**
Summary	Summary of Avatars' participation in discussions.

7.1.15.4 JSON Metadata

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

7.1.15.5 Conformance Testing

Table 2 provides the Conformance Testing Method for MMC-SCM) 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-SCM AIM

Input	Entity ID	Shall validate against Instance ID schema.
	Text Object	Shall validate against Text Object schema.
	Space-Time	Shall validate against Space-Time schema.

	Personal Status	Shall validate against Personal Status schema.
	Edited Summary	Shall validate against Summary schema.
Output	Summary	Shall validate against Summary schema.

7.1.16 Speaker Identity Recognition

7.1.16.1 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.

7.1.16.2 Reference Model

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

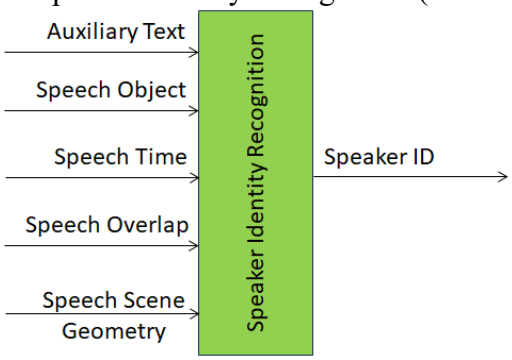

Figure 1 – The Speaker Identity Recognition (MMC-SIR) AIM

7.1.16.3 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.

7.1.16.4 JSON Metadata

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

7.1.16.5 Reference Software

7.1.16.5.1 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.

7.1.16.5.2 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

7.1.16.5.3 Acknowledgements

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

7.1.16.6 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.

7.1.16.7 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.

7.1.17 Speech Personal Status Extraction

7.1.17.1 Functions

The Speech Personal Status Extraction (MMC-SPE) AIM receives an utterance (input speech) or its Speech Descriptors and produces the Speech Personal Status using the input speech or its descriptors based on the value of input selector:

Receives	*Input Selector*	signaling whether a Speech Object or Speech Descriptors are provided.
	Speech Object	from which the Personal Status should be extracted.
	Speech Descriptors	from which the Personal Status should be extracted. SPD are externally computed
Produces	*Speech Personal Status.*	Entity's Personal Status.

7.1.17.2 Reference Model

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

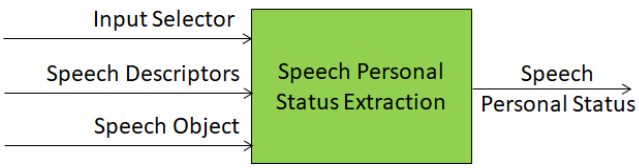

Figure 1 – Speech Personal Status Extraction (MMC-SPE) AIM Reference Model

7.1.17.3 Input/Output Data

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

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

Input	Description
Input Selector	Signals whether a Speech Object or its Speech Descriptors should be used to extracts Speech Personal Status .
Speech Object	Speech Object from which the Personal Status should be extracted..
Speech Descriptors	Descriptors from which the Personal Status should be extracted..
Output	**Description**
Speech Personal Status	The computed Speech Personal Status.

7.1.17.4 SubAIMs

A Speech Personal Status Extraction (MMC-SPE) AIM instance may be implemented as a Composite AIM as specified in Figure 2.

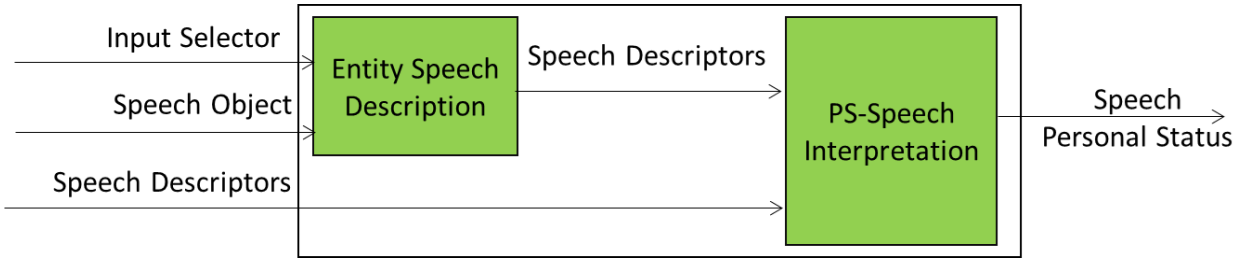

Figure 2 - Reference Model of Speech Personal Status Extraction (MMC-SPE) Composite AIM

7.1.17.5 JSON Metadata

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

7.1.17.6 Conformance Testing

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

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

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

Table 3 provides an example of MMC-SPEAIM conformance testing.

Table 3 - An example MMC-AQM SPE conformance testing

Input Data	**Data Type**	**Input Conformance Testing Data**
Speech Object	.wav	All input Speech files to be drawn from Speech files.
Output Data	**Data Type**	**Data Format**
Emotion (Speech)	JSON	All Emotion JSON files shall validate against Emotion Schema.

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

7.1.18 Speech Translation with Descriptors

7.1.18.1 Functions

The Speech Translation with Descriptors (MMC-STD) AIM receives an utterance (input speech) and a code identifying a language and produces an output speech which is a translation of the input speech in that language with the speech descriptors of the input speech:

Receives	Speech Object
	Language Selector
Produces	Synthesised Translated Speech Object having the Descriptors of the input Speech Object.

7.1.18.2 Reference Model

Figure 1 depicts the Reference Model of the Speech Translation with Descriptors (MMC-STD) AIM.

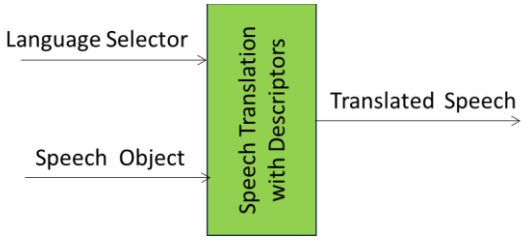

Figure 1 – The Speech Translation with Descriptors (MMC-STD) AIM Reference Model

7.1.18.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Speech Translation with Descriptors (MMC-STD) AIM.

Table 1 – I/O Data of the Speech Translation with Descriptors (MMC-STD) AIM

Input	Description
Speech Object	Input Speech.
Language Selector	Provides codes of the input and output languages.
Output	**Description**
Speech Object	Output Speech of the Text-To-Speech AIM,

7.1.18.4 SubAIMs

Speech Translation with Descriptors may also be implemented as a Composite AIM as specified in Figure 2.

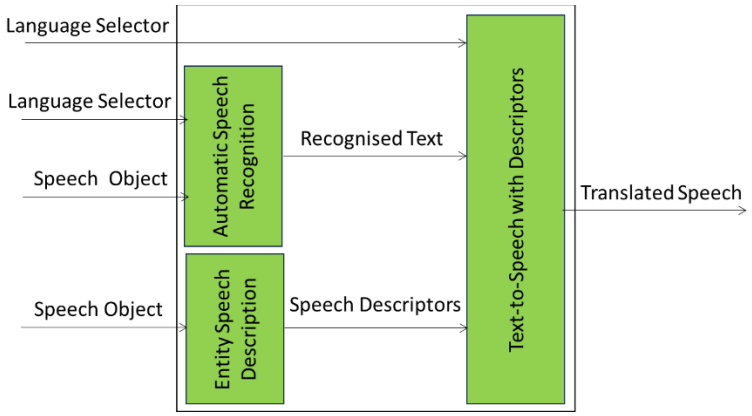

Figure 2 - Speech Translation with Descriptors (MMC-STD) Composite AIM

Table 2 specifies the AIMs of Speech Translation with Descriptors (MMC-STD) Composite AIM

Table 2 - AIMs of Speech Translation with Descriptors (MMC-STD) Composite AIM

AIM		Name	JSON
MMC-DST		Speech Translation with Descriptors	X
	MMC-ASR	Automatic Speech Recognition	X
	MMC-ESD	Entity Speech Description	X
	MMC-SDT	Descriptors Text Translation	X

7.1.18.5 JSON Metadata

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

7.1.18.6 Conformance Testing

Table 2 provides the Conformance Testing Method for MMC-STD AIM. Conformance Testing of the individual AIMs of the MMC-STD 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 MMC-STD AIM

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

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

Table 3 – An example MMC-STD AIM conformance testing

Input Data	**Data Type**	**Input Conformance Testing Data**
Language Selector	Selector	All Language Selectors to be drawn from Language Codes.
Speech Object	Speech	All input Speech files to be drawn from Speech files.
Output Data	**Data Type**	**Input Conformance Test Results**
Translated Speech	Speech	All Speech files produced shall conform with Speech files.

7.1.19 Text-to-Speech with Descriptors

7.1.19.1 Functions

The Text-to-Speech with Descriptors (MMC-TSD) AIM receives an input text and Speech Descriptors and produces an output speech that is a synthetic version of the text uttered with the input Speech Descriptors:

Receives	Text Object	to be translated with the colour of the input Speech Descriptors.
	Speech Descriptors	to be used to produce synthetic Speech.

Produces	Synthesised Speech Object	having the Descriptors of the input Speech Object.

7.1.19.2 Reference Model

Figure 1 depicts the Reference Model of the Text-to-Speech with Descriptors (MMC-TSD) AIM.

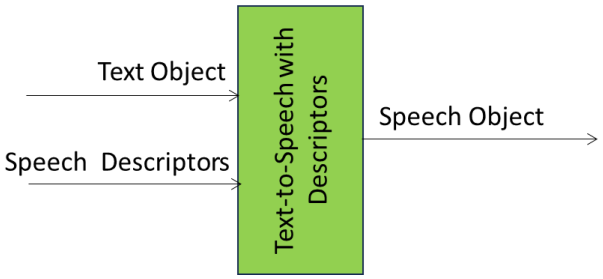

Figure 1 – The Text-to-Speech with Descriptors (MMC-TSD) AIM Reference Model

7.1.19.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Text-to-Speech with Descriptors (MMC-TSD) AIM.

Table 1 – I/O Data of the Text-to-Speech with Descriptors (MMC-TSD)AIM

Input	Description
Text Object	Input Text to be translated as Speech.
Speech Descriptors	The set of input Speech features.
Output	**Description**
Speech Object	Output Speech of the Descriptors Text-to-Speech (MMC-DTS) AIM.

7.1.19.4 JSON Metadata

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

7.1.19.5 Conformance Testing

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

Input	Text Object	Shall validate against Text Object schema. Text Data shall conform with Text Qualifier

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

7.1.20 Text and Speech Translation

7.1.20.1 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.

7.1.20.2 Reference Model

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

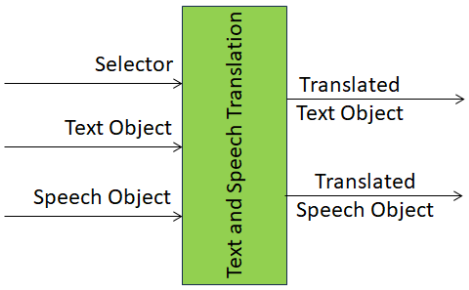

Figure 1 – Text-and-Speech Translation (MMC-TST) AIM Reference Model

7.1.20.3 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.

7.1.20.4 SubAIMs

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

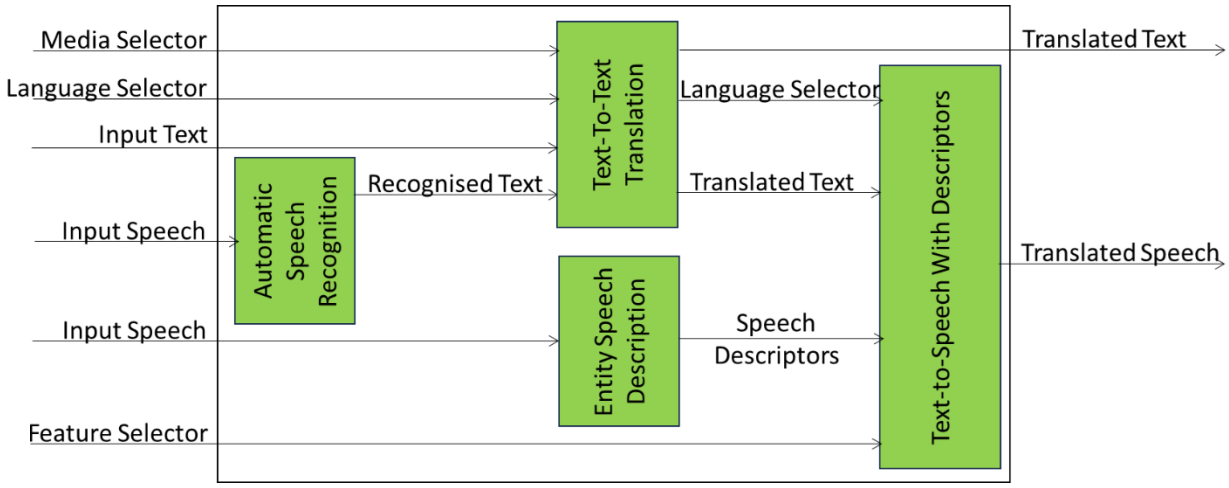

Figure 2 – Text-and-Speech Translation Composite (MMC-TST) AIM

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

7.1.20.5 JSON Metadata

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

7.1.20.6 Profiles

The Profiles of Text and Speech Translation are specified.

7.1.20.7 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.

7.1.21 Text and Image Query

7.1.21.1 Functions

The Text and Image Query (MMC-TIQ) AIM receives an input text and an input image and produces an output text that is a response to the inputs:

Receives	Text Object	Textual part of query.
	Image Visual Object	Image part of query.
Produces	Text Object	In response to Text and Image provided as input.

7.1.21.2 Reference Model

Figure 1 depicts the Reference Model of the Text and Image Query (MMC-TIQ) AIM.

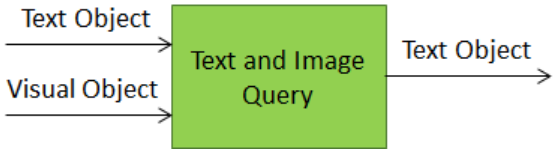

Figure 1 – The Text and Image Query (MMC-TIQ) AIM Reference Model

7.1.21.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Text and Image Query (MMC-TIQ) AIM.

Table 1 – I/O Data of the Text and Image Query (MMC-TIQ) AIM

Input	Description
Text Object	Text asking question about the Image.
Visual Object	Image about which a question is asked.
Output	**Description**
Text Object	Response produced by Text and Image Query.

7.1.21.4 SubAIMs

Text and Image Query (MMC-TIQ) can be implemented as a Composite AIM whose Reference Model is depicted in Figure 2.

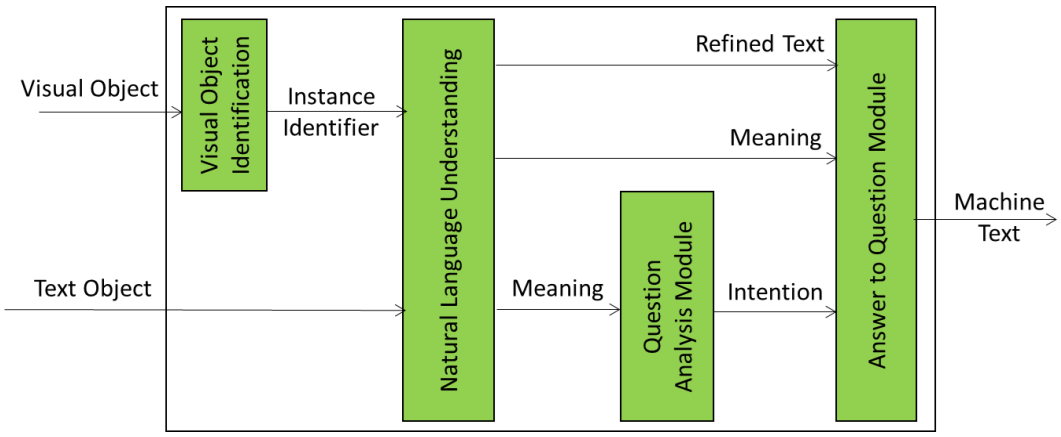

Figure 2 - Text and Image Query (MMC-TIQ) Composite AIM Reference Model

The AIMs and there JSON Metadata are specified in Table 2

Table 2 – AIMs and JSON Metadata of Text and Image Query (MMC-TIQ) Composite AIM

Acronym		AIM Name	JSON
MMC-TIQ		Text-and-Image Query	X
	OSD-VOI	Visual Object Identification	X
	MMC-NLU	Natural Language Understanding	X
	MMC-QAM	Question Analysis Module	X
	MMC-AQM	Answer to Question Module	X

7.1.21.5 JSON Metadata

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

7.1.21.6 Reference Software

7.1.21.6.1 Disclaimers

1. The purpose of this MMC-TIQ Reference Software is to provide a working Implementation of MMC-TIQ, 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.

7.1.21.6.2 Guide to the TIQ code

Note that the Reference software implements the Basic MMC-TIQ AIM.
Use of this AI Module is for developers who are familiar with Python and downloading models from HuggingFace,
A wrapper for the BLIP NN Module:
1. Manages input files and parameters: Text Object, Visual Object
2. Executes the BLIP Module to perform the question answering on each individual pair of Text and Visual Object.
3. Outputs Text Object as answer.

The OSD-TIQ Reference Software is found at the NNW gitlab site. It contains:
1. The python code implementing the AIM.
2. Required libraries are: pytorch and transformers (HuggingFace), PIL

7.1.21.6.3 Acknowledgements

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

7.1.21.7 Conformance Testing

The Conformance Testing Method for the MMC-TIQ Basic AIM is provided here. The Conformance Testing Methods for the individual Basic AIMs of the MMC-TIQ Composite AIM are provided by the individual Basic AIMs.
Table 3 provides the Conformance Testing Method for MMC-TIQ 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 3 – Conformance Testing Method for MMC-TIQ AIM

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

7.1.22 Text-To-Speech

7.1.22.1 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.

7.1.22.2 Reference Model

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

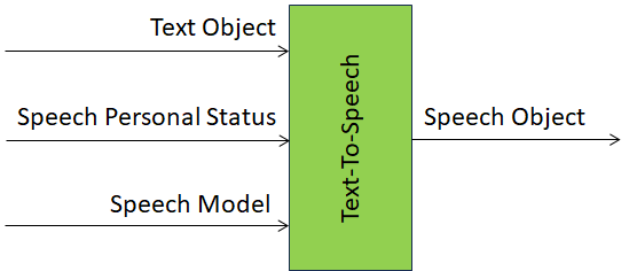

Figure 1 – The Text-To-Speech AIM Reference Model

7.1.22.3 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,

7.1.22.4 JSON Metadata

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

7.1.22.5 Profiles

The Text-To-Speech Profiles are specified.

7.1.22.6 Reference Software

7.1.22.6.1 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.

7.1.22.6.2 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.

7.1.22.6.3 Acknowledgements

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

7.1.22.7 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

Input Data	**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.

7.1.23 Text-to-Text Translation

7.1.23.1 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.

7.1.23.2 Reference Model

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

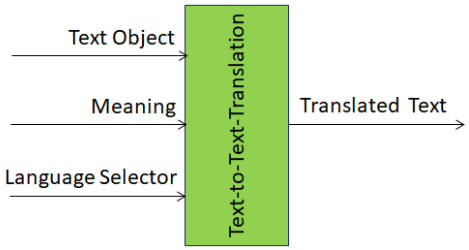

Figure 1 – Text-to-Text Translation AIM Reference Model

7.1.23.3 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).

7.1.23.4 JSON Metadata

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

7.1.23.5 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.

7.1.23.6 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.

7.1.24 Video Lip Animation

7.1.24.1 Functions

The Video Lip Animation (MMC-VLA) AIM receives an input speech and an Emotion, it queries a video from a KB and produces a Video moving lips in sync with the text and displaying the input Emotion:

Receives	Speech Object	E.g., from upstream AIM.
	Machine Emotion	From upstream AIM.
	Video	From the KB of Videos of Faces.
Produces	Face Object	Displaying lips animated by Speech on a face Emotion.

7.1.24.2 Reference Model

Figure 1 depicts the Reference Model of the Video Lip Animation (MMC-VLA) AIM.

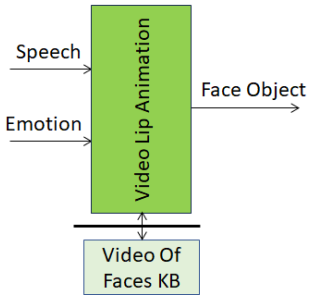

Figure 1 – The Video Lip Animation (MMC-VLA) AIM Reference Model

7.1.24.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Video Lip Animation (MMC-VLA) AIM.

Table 1 – I/O Data of the Video Lip Animation (MMC-VLA) AIM

Input data	Description
Speech Object	An input Speech that may have been produced by a Machine.
Emotion	Emotion in the input Speech.
Video Object	A video drawn from a KB of Faces.
Output data	**Description**
Emotion	Emotion in the input Speech. that is used to alter the Video drawn from the KB Faces.
Face Object	Synthetic speaking Face Object displaying Emotion.

7.1.24.4 JSON Metadata

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

7.1.24.5 Conformance Testing

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

Input	Machine Speech Object	Shall validate against Speech Object schema. Speech Data shall conform with Speech Qualifier.
	Machine Emotion	Shall validate against Emotion schema.

	Video Object	Shall validate against Visual Object schema. Visual Data shall conform with Visual Qualifier.
Output	Face Object	Shall validate against Visual Object schema. Visual Data shall conform with Visual Qualifier.

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

Table 3 – An example MMC-VLA AIM conformance testing

Input Data	Data Type	Input Conformance Testing Data
Machine Speech	.wav	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
Video of Face	AVC	All input Video files to be drawn from Video files.
Output Data	**Data Type**	**Output Conformance Testing Criteria**
Machine Face	AVC	All Video files produced shall conform with Video.

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-MMC 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-MMC AIMs have a Reference Software Implementation.

7.3 Conformance Testing

An implementation of an AI Module conforms with MPAI-MMC 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 MPAI-MMC.

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.

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.

The current MPAI-MMC V2.3 Standard does not provide AIM Performance Assessment methods.

8 Data Types

8.1 Technical Specifications

This page gives the links to the specification of Data Types specified by ***Technical Specification: Multimodal Conversation (MPAI-MMC) V2.4***. All previously specified MPAI-MMC Data Types are superseded by those specified by V2.4.

Acronym	Name	JSON	Acronym	Name	JSON
MMC-ECS	Cognitive State	X	MMC-SPD	Speech Descriptors	X
MMC-EEM	Emotion	X	MMC-SOV	Speech Overlap	X
MMC-FPS	Face Personal Status	X	MMC-SPS	Speech Personal Status	X
MMC-GPS	Gesture Personal Status	X	MMC-SUM	Summary	X
MMC-INT	Intention	X	MMC-TXD	Text Descriptors	X
MMC-MEA	Meaning	X	MMC-TPS	Text Personal Status	X
MMC-EPS	Personal Status	X	MMC-TXS	Text Segment	X
MMC-ESC	Social Attitude	X	MMC-TXW	Text Word	X

8.1.1 Cognitive State

8.1.1.1 Definition

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

8.1.1.2 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.

8.1.1.3 Syntax

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

8.1.1.4 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

8.1.1.5 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.

8.1.2 Emotion

8.1.2.1 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".

8.1.2.2 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.

8.1.2.3 Syntax

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

8.1.2.4 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

8.1.2.5 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.

8.1.3 Face Personal Status

8.1.3.1 Definition

Face Personal Status is a Data Type including the 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").

of an Entity's Face Modality.

8.1.3.2 Functional Requirements

Face Personal Status is added for convenience. However, it is simply the Personal Status of the Face Modality.

8.1.3.3 Syntax

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

8.1.3.4 Semantics

Label	Description
Header	Header of Face Personal Status
- Standard	The characters "MMC-FPS-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
FacePersonalStatusID	Identifier of Face Personal Status.
FacePersonalStatusSpaceTime	Space-Time info of Face Personal Status
FacePersonalStatus	Face Personal Status
- FaceCognitiveState	Cognitive State component of Face Personal Status

- FaceEmotion	Emotion component of Face Personal Status
- FaceSocialAttitude	Social Attitude component of Face Personal Status
DescrMetadata	Descriptive Metadata

8.1.3.5 Conformance Testing

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

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

8.1.4 Gesture Personal Status

8.1.4.1 Definition

Gesture Personal Status is a Data Type including the 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").

of an Entity's Gesture Modality.

8.1.4.2 Functional Requirements

Gesture Personal Status is added for convenience. However, it is simply the Personal Status of the Gesture Modality.

8.1.4.3 Syntax

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

8.1.4.4 Semantics

Label	Description
Header	Header of Gesture Personal Status
- Standard	The characters "MMC-GPS-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
GesturePersonalStatusID	Identifier of Gesture Personal Status.
GesturePersonalStatusSpaceTime	Space-Time info of Gesture Personal Status
GesturePersonalStatus	Gesture Personal Status
- GestureCognitiveState	Cognitive State component of Gesture Personal Status
- GestureEmotion	Emotion component of Gesture Personal Status

- GestureSocialAttitude	Social Attitude component of Gesture Personal Status
DescrMetadata	Descriptive Metadata

8.1.4.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V1.3 (MMC-GPS) if:

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

8.1.5 Intention

8.1.5.1 Definition

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

8.1.5.2 Functional Requirements

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

8.1.5.3 Syntax

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

8.1.5.4 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

8.1.5.5 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.

8.1.6 Meaning

8.1.6.1 Definition

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

8.1.6.2 Functional Requirements

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

8.1.6.3 Syntax

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

8.1.6.4 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

8.1.6.5 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.

8.1.7 Personal Status

8.1.7.1 Definition

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

8.1.7.2 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.

8.1.7.3 Syntax

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

8.1.7.4 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

8.1.7.5 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.

8.1.8 Social Attitude

8.1.8.1 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"..

8.1.8.2 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	
APPROVAL/DISAPPROVAL	admiring/approving	awed
		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.

8.1.8.3 Syntax

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

8.1.8.4 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.

8.1.8.5 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.

8.1.9 Speech Descriptors

8.1.9.1 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.

8.1.9.2 Functional Requirements

Speech Descriptors may include Neural Network Descriptors.

8.1.9.3 Syntax

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

8.1.9.4 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.

8.1.9.5 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.

8.1.10 Speech Overlap

8.1.10.1 Definition

A Data Type representing information of a Speech Object overlapping utterances and their duration.

8.1.10.2 Functional Requirements

The Speech Overlap Data Type include a set of:

1. IDs of overlapping utterances.
2. Duration of utterances.

8.1.10.3 Syntax

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

8.1.10.4 Semantics

Label	Description
Header	Speech Overlap Header
- Standard	The characters "MMM-SOL-V"
- Version	Major version – 1 or 2 Bytes
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 Bytes
M-InstanceID	Identifier of M-Instance.
SpeechOverlapID	Identifier of Speech Overlap.
SpeechQualifier	Qualifier of overlapping Speech Objects.
SpeechOverlapData[]	The set of Contract Data
- SpeechObjectID	ID of Speech Object.
- SpeechObjectTime	Time of Speech Object.
DescrMetadata	Descriptive Metadata

8.1.10.5 Conformance Testing

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

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

8.1.11 Speech Personal Status

8.1.11.1 Definition

Speech Personal Status is a Data Type including the 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").

of an Entity's Speech Modality.

8.1.11.2 Functional Requirements

Speech Personal Status is added for convenience. However, it is simply the Personal Status of the Speech Modality.

8.1.11.3 Syntax

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

8.1.11.4 Semantics

Label	Description
Header	Header of Speech Personal Status
- Standard	The characters "MMC-SPS-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SpeechPersonalStatusID	Identifier of Speech Personal Status.
SpeechPersonalStatusSpaceTime	Space-Time info of Speech Personal Status
SpeechPersonalStatus	Speech Personal Status
- SpeechCognitiveState	Cognitive State component of Speech Personal Status
- SpeechEmotion	Emotion component of Speech Personal Status
- SpeechSocialAttitude	Social Attitude component of Speech Personal Status
DescrMetadata	Descriptive Metadata

8.1.11.5 Conformance Testing

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

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

8.1.12 Summary

8.1.12.1 Definition

A Data Type representing a text-based abridged outline of the utterance(s) of one or more Entities represented by their User ID and including Space-Time, Text, and Personal Statuses.

8.1.12.2 Functional Requirements

Summary includes:

1. Virtual Space where Summary was generated (M-Instance).
2. Space-Time information in the Virtual Instance.
3. Content for each speaking Entity:
    1. Text
    2. Space-Time information of the Entity the Text refers to.
    3. Personal Status of the Entity the Text refers to.

8.1.12.3 Syntax

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

8.1.12.4 Semantics

Label	Description
Header	Summary Header
- Standard-Item	The characters “MMC-SUM-V”
- Version	Major version – 1 or 2 Bytes
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 Bytes
MInstanceID	Identifier of M-Instance.
SummaryID	Identifier of the Summary.
SummarySpaceTime	Space-Time of Summary.
SummaryData[]	Data of Summary
- ReportedEntityID	ID of the Entity Reported in Summary
- ReportedEntityPersonalStatus	Personal Status of Entity Reported in Summary
- ReportedEntitySpaceTime	Time-Space info of Entity Reported in Summary
- ReportedEntityTextObject	Text Object of Entity Reported in Summary
SummaryData	Summary Data.
- SummaryDataLength	Number of Bytes in Summary Data
- SummaryDataURI	URI of Data of Summary Data
DescrMetadata	Descriptive Metadata

8.1.12.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.4 Summary (MMC-SUM) if:

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

8.1.13 Text Descriptors

8.1.13.1 Definition

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

8.1.13.2 Functional Requirements

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

8.1.13.3 Syntax

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

8.1.13.4 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

8.1.13.5 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.

8.1.14 Text Personal Status

8.1.14.1 Definition

Text Personal Status is a Data Type including the 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").

of an Entity's Text Modality.

8.1.14.2 Functional Requirements

Text Personal Status is added for convenience. However, it is simply the Personal Status of the Text Modality.

8.1.14.3 Syntax

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

8.1.14.4 Semantics

Label	Description
Header	Header of Text Personal Status
- Standard	The characters "MMC-TPS-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
TextPersonalStatusID	Identifier of Text Personal Status.
TextPersonalStatusSpaceTime	Space-Time info of Text Personal Status
TextPersonalStatus	Text Personal Status
- TextCognitiveState	Cognitive State component of Text Personal Status
- TextEmotion	Emotion component of Text Personal Status
- TextSocialAttitude	Social Attitude component of Text Personal Status
DescrMetadata	Descriptive Metadata

8.1.15 Text Segment

8.1.15.1 Definition

A Data Type consisting of Text Words separated by spaces, typically of a limited length.

8.1.15.2 Functional Requirements

1. A Text Segment may include the Time of the start and the end of the speech segment.
2. When the Text Segment is the output of an ASR Implementation, a Confidence Score may be attached to the Text Segment.
3. The Confidence Score is a number between 0 and 1.

8.1.15.3 Syntax

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

8.1.15.4 Semantics

Label	Description
Header	Text Segment Header
– Standard-TextSegment	The characters "MMC-TXS-V"
– Version	Major version – 1 or 2 characters
– Dot-separator	The character "."
– Subversion	Minor version – 1 or 2 characters

MInstanceID	Identifier of M-Instance.
Time	Time of the start of Text Segment.
TextSegmentID	Identifier of Text Segment.
ConfidenceScore	The Confidence of the ASR in the correctness of the Text Segment.
TextSegmentData[]	Data of Text Segment.
– TextWord	Text represented by a string.
DescrMetadata	Descriptive Metadata

8.1.15.5 Conformance Testing

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

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

8.1.16 Text Word

8.1.16.1 Definition

A Data Type consisting of characters, typically of a limited length.

8.1.16.2 Functional Requirements

1. When the Text Word is the output of an ASR, the Text Segment may include:
    1. a Confidence Score.
    2. A Time stamp indicating the times the word starts and ends.
2. The Confidence Score is a number comprised between 0 and 1.

8.1.16.3 Syntax

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

8.1.16.4 Semantics

Label	Description
Header	Text Word Header
– Standard-TextWord	The characters "MMC-TXW-V"
– Version	Major version – 1 or 2 characters
– Dot-separator	The character "."
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
TextWordID	Identifier of Text Segment.
TextWordConfidenceScore	The Confidence of the ASR in the correctness of the Text Word.
TextWordData	Data of Text Word.
– String	String representing the Word.
– Time	Time indicating the start and end of the Word.
DescrMetadata	Descriptive Metadata

8.1.16.5 Conformance Testing

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

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

8.2 Conformance testing

A Data instance of a Data Type specified by MPAI-MMC V2.4 Conforms with it if the JSON Data validate against the relevant MPAI-MMC V2.4 JSON Schema and if the Data Conforms with the relevant Data Qualifier, if present. MPAI-MMC V2.4 does not provide method for testing the Conformance of the Semantics of the Data instance to the MPAI-MMC V2.4 specification.

Conformance testing can be performed by a human using a JSON Validator to verify the Conformance of the syntax of JSON Data to the relevant JSON Schema; and, if the Data has a Qualifier, to verify that the syntax of the Data conforms with the relevant values in the Data Qualifier. Alternatively, Conformance testing can be performed by software implementing the steps above.

8.3 Performance Assessment

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

1. *Quality*: Performance Assessment measures the quality of the Data instance using a metric that depends on the nature of the Data, e.g., the word error rate (WER) of a string of characters representing a sentence compared to an idea sentence.
2. *Bias*: Performance Assessment uses a metric that depends on the bias in the Data compared with reference Data related to certain attributes of the Data. For instance, the Data may contain information about a particular geographic area when the ideal data do not .
3. Legal compliance: Performance Assessment uses an appropriate metric to measure how well the Data instance complies with with a certain legal standard.

9 Datasets

9.1 Introduction

Testing the Conformance of MMC-CWE, MMC-MQA, and MMC-UST requires datasets to test Data, AIMs, and AIWs. The Data Formats belong to one of Text, Audio, Video, and JSON and should have the characteristics of Table 1:

Table 1 – Data Types for Conformance Testing of MMC-CWE, MMC-MQA, and MMC-UST

Data Type	Characteristics
Text	The texts files shall be composed of Unicode characters.
Speech file	The speech files shall be conforming .wav files.
Video file	The video files shall be conforming MP4 files.
Image File	The Image file shall be conforming
Emotion	Emotion files shall be JSON files conforming with the Emotion JSON Schema.

Intention	Emotion files shall be JSON files conforming with the Intention JSON Schema.
Meaning	Emotion files shall be JSON files conforming with the Meaning JSON Schema.

Humans shall carry out Conformance Testing by visual and auditory inspection. Appropriate software may replace humans as Conformance Testers.
Conformance Testing Datasets are publicly available upon registration.

9.2 Text with Emotion

9.2.1 Coherent scenarios

Happy	1. Today was a wonderful day. I spent quality time with my parents, and the restaurant was excellent as well. I look forward to seeing them again! 2. I'm so excited about Christmas. This year, my girlfriend and I are going to celebrate the holiday together. We'll decorate our room, and it'll be so much fun. 3. Today I watched a movie called 'The Pianist.' Not only was it touching, but also very absorbing. Now I feel very happy thanks to the memorable experience. 4. The weather is awesome these days. It is not too cold, not too hot, and the sun shines beautifully. I look forward to the picnic that is scheduled this weekend. 5. Nowadays my business is running very smoothly. There are no unexpected issues arising, and my employees are working very diligently. I am very relieved.
Angry	1. Today my coworkers treated me really badly. They blamed me for the things that were neither my responsibility nor the result of my actions. This is so unfair. 2. I am angry with my sister. She not only does not finish her chores, but forces me to do the chores for her. This is not a new occasion, but this time I can't, stand it. 3. Yesterday I had an argument with a friend of mine. He always wants me to listen to him very carefully and provide advice, but when I'm in need of the help of the same sort, he doesn't fulfill his duty at all. I'm furious about this. 4. These days consumer price is skyrocketing. However, the government and political parties are busy blaming the external variables, not trying hard to solve. the problem that ordinary citizens are facing. Why is there no one trying to be responsible? 5. Because of my superior in my workplace, I am doing monotonous tasks all day long these days. I have to look at thousands of boring images and classify them each day, which drives me crazy. I cannot but blame my superior.
Neutral	1. Seoul is the capital city of the Republic of Korea. It is a city of almost ten million residents. According to "The Global Livability Index" Seoul is ranked the fourth most livable city in Asia as of 2023. 2. There is a famous proverb, "Honesty is the best policy." In essence, it suggests that honesty is the most effective and beneficial approach in various aspects of life. 3. There is a famous saying, "Don't judge a book by its cover." This advises people not to form an opinion or make assumptions about someone or something based solely on its outward appearance. 4. Global warming refers to the long-term increase in Earth's average surface temperature due to human activities, primarily the emission of greenhouse gases. Greenhouse gases trap heat in the Earth's atmosphere, leading to the warming effect.

5. Inflation is a general increase of the prices of goods and services in an economy. This is usually measured using the consumer price index (CPI).

9.2.2 Incoherent scenarios

Text	Meaning	Speech	Face	Sentences
Happy	Happy	Angry	Angry	I'm headed to a yoga class now, and then I have a cozy evening planned with a good book. Life is good, for sure.
Happy	Happy	Neutral	Neutral	With a big scoop of ice cream in hand, I laughed and played in the park, feeling super happy as the sun shone brightly overhead.
Angry	Angry	Happy	Happy	Witnessing my neighbor being rude and disrespectful to an old stranger asking for directions, I couldn't be sane, because that old man was my father.
Neutral	Neutral	Happy	Happy	A political party is an organization that coordinates candidates to compete in a particular country's elections. It is common for the members of a party to hold similar ideas about politics.
Neutral	Neutral	Angry	Angry	According to Max Weber, a state is a compulsory political organization with a centralized government that maintains a monopoly of the legitimate use of force within a certain territory.

9.3 Audio and Video with Emotion

9.3.1 Neutral

MPAI emotions neutral 1 audio.240309.1041.wav
MPAI emotions neutral 1 video.240309.1041.mp4
MPAI emotions neutral 1.240309.1041.mp4
MPAI emotions neutral 2 audio.240309.1041.wav
MPAI emotions neutral 2 video.240309.1041.mp4
MPAI emotions neutral 2.240309.1041.mp4
MPAI emotions neutral 3 audio.240309.1041.wav
MPAI emotions neutral 3 video.240309.1041.mp4
MPAI emotions neutral 3.240309.1041.mp4
MPAI emotions neutral 4 audio.240309.1041.wav
MPAI emotions neutral 4 video.240309.1041.mp4
MPAI emotions neutral 4.240309.1041.mp4
MPAI emotions neutral 5 audio.240309.1041.wav
MPAI emotions neutral 5 video.240309.1041.mp4
MPAI emotions neutral 5.240309.1041.mp4

9.3.2 Angry

MPAI emotions angry 5.240309.1041.mp4
MPAI emotions angry 5 video.240309.1041.mp4
MPAI emotions angry 5 audio.240309.1041.wav
MPAI emotions angry 4.240309.1041.mp4
MPAI emotions angry 4 video.240309.1041.mp4
MPAI emotions angry 4 audio.240309.1041.wav

MPAI emotions angry 3.240309.1041.mp4
MPAI emotions angry 3 video.240309.1041.mp4
MPAI emotions angry 3 audio.240309.1041.wav
MPAI emotions angry 2.240309.1041.mp4
MPAI emotions angry 2 video.240309.1041.mp4
MPAI emotions angry 2 audio.240309.1041.wav
MPAI emotions angry 1.240309.1041.mp4
MPAI emotions angry 1 video.240309.1041.mp4
MPAI emotions angry 1 audio.240309.1041.wav

9.3.3 Happy

MPAI emotions happy 1 audio.240309.1041.wav
MPAI emotions happy 1 video.240309.1041.mp4
MPAI emotions happy 1.240309.1041.mp4
MPAI emotions happy 2 audio.240309.1041.wav
MPAI emotions happy 2 video.240309.1041.mp4
MPAI emotions happy 2.240309.1041.mp4
MPAI emotions happy 3 audio.240309.1041.wav
MPAI emotions happy 3 video.240309.1041.mp4
MPAI emotions happy 3.240309.1041.mp4
MPAI emotions happy 4 audio.240309.1041.wav
MPAI emotions happy 4 video.240309.1041.mp4
MPAI emotions happy 4.240309.1041.mp4
MPAI emotions happy 5 audio.240309.1041.wav
MPAI emotions happy 5 video.240309.1041.mp4
MPAI emotions happy 5.240309.1041.mp4

9.3.4 Incoherent

MPAI emotions angry text happy voice.240309.1041.mp4
MPAI emotions angry text happy voice audio.240309.1041.wav
MPAI emotions angry text happy voice video.240309.1041.mp4
MPAI emotions happy text angry voice.240309.1041.mp4
MPAI emotions happy text angry voice audio.240309.1041.wav
MPAI emotions happy text angry voice video.240309.1041.mp4
MPAI emotions happy text neutral voice.240309.1041.mp4
MPAI emotions happy text neutral voice audio.240309.1041.wav
MPAI emotions happy text neutral voice video.240309.1041.mp4
MPAI emotions neutral text angry voice.240311.0915.mp4
MPAI emotions neutral text angry voice audio.240311.0915.wav
MPAI emotions neutral text angry voice video.240311.0915.mp4
MPAI emotions neutral text happy voice audio.240309.1041.wav
MPAI emotions neutral text happy voice video.240309.1041.mp4

9.4 Emotion JSON Files

The JSON files below represent Happy, Angry, and Neutral Emotions.


```
{
  "EmotionType":{
    "emotionDegree":"high",
    "emotionName":"happy",
    "emotionSetName":"MPAI Basic Emotion Set"
```

```
  }
}

{
  "EmotionType":{
    "emotionDegree":"high",
    "emotionName":"happy",
    "emotionSetName":"MPAI Basic Emotion Set"
  }
}

{
  "EmotionType":{
    "emotionDegree":"high",
    "emotionName":"happy",
    "emotionSetName":"MPAI Basic Emotion Set"
  }
}
```

9.5 Meaning JSON Files

Sentence 1: Today was a wonderful day! I spent quality time with my parents, and the McDonald restaurant was excellent, too. I'm looking forward to seeing them again!

```
{
  "meaning": {
    "POS_tagging": {
      "POS_tagging_set": "CST's Part-Of-Speech tagger (Brill, with
adaptations)",
      "POS_tagging_result": "Today/RB was/VBD a/DT wonderful/JJ
day/NN !/. I/PRP spent/VBD quality/NN time/NN with/IN
my/PRP$ parents/NNS ,/, and/CC the/DT McDonald/NNP     restaurant/NN
was/VBD excellent/JJ ,/, too/RB ./. I'm/NNP looking/VBG forward/RB
to/TO seeing/VBG them/PRP again/RB !/."
    },
    "NE_tagging": {
      "NE_tagging_set": "CST's named entity recogniser",
      "NE_tagging_result": " [Today,misc,uncertain] was a wonderful
day ! I spent quality time with my parents, and the
[McDonald,person,likely] restaurant was excellent , too . I'm looking
forward to seeing them again!"
    },
    "dependency_tagging": {
      "dependency_tagging_set": "CG-dependency,
https://edu.visl.dk/visl/en/parsing/automatic/dependency.php ",
      "dependency_tagging_result": "<ß>nToday [today] <*> <atemp> ADV
@ADVL> #1->2nwas [be] <mv> V IMPF 1/3S @FS-STA #2->0na [a] <indef> ART
S @>N #3->5nwonderful [wonderful] ADJ POS @>N #4->5nday [day] <dur>
<per> <idf> <nhead> N S NOM @<SUBJ #5->2n! [!] PU @PU #6->0n</s>n<ß>nI
[I] <*> PERS 1S NOM @SUBJ> #1->2nspent [spend] <cjt-head> <mv> V IMPF
@FS-STA #2->0nquality [quality] <f-q> <f-phys> <comp1> <first> <idf>
<comp1> <ncomp> N S NOM @>N #3->4ntime [time] <ac-cat> <temp> <per>
<num+> <second> <comp2> <idf> <nhead> N S NOM @<ACC #4->2nwith [with]
PRP @<ADVL #5->2nmy [I] <poss> <refl> <det> PERS 1S GEN @>N
#6->7nparents [parent] <Hfam> <def> <nhead> N P NOM @P< #7->5n, [,] PU
@PU #8->0nand [and] <clb?> <co-fin> KC @CO #9->2nthe [the] <def> ART
```

```
S/P @>N #10->12nMcDonald [McDonald] <*> <Proper> <first> <ncomp> N S
NOM @>N #11->12nrestaurant [restaurant] <inst> <second> <def> <nhead>
N S NOM @SUBJ> #12->13nwas [be] <cjt> <mv> V IMPF 1/3S @FS-STA
#13->2nexcellent [excellent] <Q:good> ADJ POS @<SC #14->13n, [,] PU
@PU #15->0ntoo [too] ADV @<ADVL #16->13n. [.] PU @PU
#17->0n</s>n<ß>nI-m [I-m] <*> <unit> <ac-sign> <heur> <idf> <nhead> N
S NOM @SUBJ> #1->2nlooking [look] <mv> V PCP1 @ICL-ADVL #2->0nforward
[forward] <adir> <advl-close> ADV @<ADVL #3->2nto [to] <advl-close>
PRP @<ADVL #4->2nseeing [see] <vq> <v.contact> <vtk+ADJ> <mv> V PCP1
@ICL-P< #5->4nthem [they] PERS 3P ACC @<ACC #6->5nagain [again]
<atemp> ADV @<ADVL #7->5n! [!] PU @PU #8->0n</ß>"
    },
    "SRL_tagging": {
      "SRL_tagging_set": "HanLP,
https://hanlp.hankcs.com/en/demos/srl.html",
      "SRL_tagging_result": "Today/ARG1 was/PRED (a wonderful
day)/ARG2 ! I/ARG0 spent/PRED (quality time)/ARG1 (with my
parents)/ARG2, and (the McDonald restaurant)/ARG1 was/PRED
excellent/ARG2, too/ARGM-ADV. I/ARG0'm looking/PRED forward/ARGM-DIR
(to seeing them again)/ARG1!"
    }
  }
}
```


Sentence 2: I'm really excited about Christmas! This year, my girlfriend and I are gonna celebrate the holiday together. We're gonna decorate our room, and it'll be so much fun!


```
{
  "meaning": {
    "POS_tagging": {
      "POS_tagging_set": "CST's Part-Of-Speech tagger (Brill, with
adaptations)",
      "POS_tagging_result": " I'm/NNP really/RB excited/VBD about/IN
Christmas/NNP !/.nThis/DT year/NN ,/, my/PRP$ girlfriend/NN and/CC
I/PRP are/VBP gon/VBG na/TO celebrate/VB the/DT holiday/NN
together/RB ./. We're/NNP gon/VBG na/TO decorate/VB
our/PRP$ room/NN ,/, and/CC it'll/NN be/VB so/RB much/JJ fun/NN !/."
    },
    "NE_tagging": {
      "NE_tagging_set": "HanLP,
https://hanlp.hankcs.com/en/demos/srl.html",
      "NE_tagging_result": " I'm really excited about Christmas/DATE !
This year, my girlfriend and I are gonna celebrate the holiday
together. We're gonna decorate our room, and it'll be so much fun! "
    },
    "dependency_tagging": {
      "dependency_tagging_set": "CG-dependency,
https://edu.visl.dk/visl/en/parsing/automatic/dependency.php ",
      " dependency_tagging_result": "n<ß>nI-m [I-m] <*> <unit> <ac-
sign> <heur> <idf> <nhead> N S NOM @NPHR #1->0nreally [really] <ly>
<ameta> <ADJ:real+ly> ADV @>A #2->3nexcited [excited] <np-close> ADJ
POS @N< #3->1nabout [about] <pp-temp> PRP @A< #4->3nChristmas
[Christmas] <*> <temp> <per> <nhead> N S NOM @P< #5->4n! [!] PU @PU
#6->0n</s>n<ß>nThis [this] <*> <dem> DET S @>N #1->2nyear [year] <per>
<dur> <def> <nhead> N S NOM @ADVL> #2->10n, [,] PU @PU #3->0nmy [I]
```

<poss> <det> PERS 1S GEN @>N #4->5ngirlfriend [girlfriend] <cjt-head> <Hfam> <def> <nhead> N S NOM @SUBJ> #5->8nand [and] <co-subj> KC @CO #6->5nI [I] <cjt> <*> PERS 1S NOM @SUBJ> #7->5nare [be] <vch> <aux> V PR -1/3S @FS-STA #8->0ngonna [going=to] <complex> <aux> V PCP1 @ICL-AUX< #9->8ncelebrate [celebrate] <mv> V INF @ICL-AUX< #10->9nthe [the] <def> ART S/P @>N #11->12nholiday [holiday] <temp> <per> <def> <nhead> N S NOM @<ACC #12->10ntogether [together] ADV @<ADVL #13->10n. [.] PU @PU #14->0n</s>n<ß>nWe-re [We-re] <*> <Hmyth> <rem> <heur> <idf> <nhead> N S NOM @SUBJ> #1->3ngonna [going=to] <cjt-head> <complex> <aux> V PCP1 @FS-STA #2->0ndecorate [decorate] <v.contact> <mv> V INF @ICL-AUX< #3->2nour [we] <poss> <det> PERS GEN 1P @>N #4->5nroom [room] <Lh> <am> <def> <nhead> N S NOM @<ACC #5->3n, [,] PU @PU #6->0nand [and] <clb?> KC @CO #7->5nit-ll [it-ll] <heur> <idf> <nhead> N S NOM @SUBJ> #8->9nbe [be] <cjt> <mv> V SUBJ @FS-STA #9->2nso [so] <aquant> ADV @>A #10->11nmuch [much] <quant> DET ABS S @>N #11->12nfun [fun] <sem-c> <percep-f> <idf> <nhead> N S NOM @<SC #12->9n! [!] PU @PU #13->0n</ß>"
    },
    "SRL_tagging": {
      "SRL_tagging_set": "HanLP, https://hanlp.hankcs.com/en/demos/srl.html",
      "SRL_tagging_result": " I/ARG1 ’m/PRED (really excited about Christmas)/ARG2! This year, (my girlfriend and I) /ARG0 are gonna celebrate/PRED (the holiday)/ARG1 together/ARGM-MNR. We/ARG0 ’re gonna decorate/PRED (our room)/ARG1, and it/ARG1 ’ll/ARGM-MOD be/PRED (so much fun)/ARG2 !"
    }
  }
}

Sentence 3: Today I watched a movie called ‘The Pianist.’ It was not only touching, but really absorbing, too. Now I’m feeling really happy, thanks to this memorable experience.


```
{
  "meaning": {
    "POS_tagging": {
      "POS_tagging_set": "CST's Part-Of-Speech tagger (Brill, with adaptations)",
      "POS_tagging_result": " Today/RB I/PRP watched/VBD a/DT movie/NN called/VBN '/'' The/DT Pianist/NNP ./. '/POS It/PRP was/VBD not/RB only/RB touching/VBG ,/, but/CC really/RB absorbing/VBG ,/, too/RB ./.nNow/RB I'm/NNP feeling/NN really/RB happy/JJ ,/, thanks/NNS to/TO this/DT memorable/JJ experience/NN ./."
    },
    "NE_tagging": {
      "NE_tagging_set": "HanLP, https://hanlp.hankcs.com/en/demos/srl.html",
      "NE_tagging_result": " Today I watched a movie called ‘The Pianist.’/WORK_OF_ART It was not only touching, but really absorbing, too. Now I’m feeling really happy, thanks to this memorable experience."
    },
    "dependency_tagging": {
      "dependency_tagging_set": "CG-dependency, https://edu.visl.dk/visl/en/parsing/automatic/dependency.php ",
```

```
      " dependency_tagging_result": "n<ß>nToday [today] <*> <atemp>
ADV @ADVL> #1->3nI [I] <*> PERS 1S NOM @SUBJ> #2->3nwatched [watch]
<DL:bio> <mv> V IMPF @FS-STA #3->0na [a] <indef> ART S @>N #4->5nmovie
[movie] <sem-w> <DL:bio> <idf> <nhead> N S NOM @<ACC #5->3ncalled
[call] <vtk+N> <vtk+ADJ> <vtk+N> <vtk+PROP> <vq> <v.contact> <DL:bio>
<mv> <np-close> V PCP2 PAS @ICL-N< #6->5n-The [-The] <heur> <DL:bio>
<idf> <nhead> N S NOM @<SC #7->6nPianist [Pianist] <*> <Proper>
<DL:bio> <nhead> N S NOM @<OC #8->6n. [.] PU @PU #9->0n<ß>n- [-] PU
@PU #1->0n</ß>n</s>n<ß>nIt [it] <*> PERS NEU 3S NOM @SUBJ> #1->2nwas
[be] <DL:bio> <mv> V IMPF 1/3S @FS-STA #2->0nnot [not] ADV @>A
#3->4nonly [only] <ly> <ADJ:on+ly> <advl-close> ADV @<ADVL
#4->2ntouching [touching] <DL:bio> ADJ POS @<SC #5->2n, [,] PU @PU
#6->0nbut [but] KC @CO #7->5nreally [really] <ly> <ameta>
<ADJ:real+ly> ADV @ADVL> #8->9nabsorbing [absorb] <v.contact> <DL:bio>
<mv> V PCP1 @ICL-N<PRED #9->1n, [,] PU @PU #10->0ntoo [too] <advl-
close> ADV @<ADVL #11->9n. [.] PU @PU #12->0n</s>n<ß>nNow [now] <*>
<atemp> ADV @ADVL #1->0nI-m [I-m] <*> <unit> <ac-sign> <DL:bio> <heur>
<nhead> N S NOM @NPHR #2->1nfeeling [feel] <v.contact> <v-cog>
<DL:bio> <mv> <np-close> V PCP1 @ICL-N<PRED #3->2nreally [really] <ly>
<ameta> <ADJ:real+ly> ADV @>A #4->5nhappy [happy] <jpsych> <DL:bio>
ADJ POS @<SC #5->3n, [,] PU @PU #6->0nthanks to [thanks=to]
<insertion> <complex> PRP @<ADVL #7->3nthis [this] <dem> DET S @>N
#8->10nmemorable [memorable] <DL:bio> ADJ POS @>N #9->10nexperience
[experience] <f-psych> <percep-f> <DL:bio> <def> <nhead> N S NOM @P<
#10->7n. [.] PU @PU #11->0n</ß>"
    },
    "SRL_tagging": {
      "SRL_tagging_set": "HanLP,
https://hanlp.hankcs.com/en/demos/srl.html",
      "SRL_tagging_result": "Today/ARG-TMP I/ARG0 watched/PRED a
movie/ARG1 called/PRED 'The Pianist.' It/ARG1 was/PRED (not only
touching, but really absorbing, too)/ARG2. Now/ARG-TMP I/ARG0 'm
feeling/PRED (really happy)/ARG1, thanks to this memorable
experience."
    }
  }
}
```


9.6 Question Text Files

Q1: What is the tool in the picture?
Q2: What is the nickname of the person in the picture?
Q3: What is the job of the person on the left hand-side in the picture
Q4: What is the family name of the person in the centre of the picture?
Q5: What is the name of the square in the picture?

9.7 Question Speech Files

Q1.wav
Q2.wav
Q3.wav
Q4.wav
Q5.wav

9.8 Images for Question

Images for Q1	Q1-1.jpg Q1-2.jpg Q1-3.jpg
Image for Q2	Q2-Joseph Gordon Levitt.jpg
Image for Q3	Q3-1.jpg Q3-2.jpg
images for Q4	Q4-1.jpg Q4-2.jpg Q4-3.jpg
1 image for Q5	Q5-1.jpg

9.9 Meaning JSON Files

Sentence 1: What is the tool in the picture?


```
{
  "meaning": {
    "POS_tagging": {
      "POS_tagging_set": "CST's Part-Of-Speech tagger (Brill, with
adaptations), https://cst.dk/online/pos_tagger/uk/",
      "POS_tagging_result": "What/WP is/VBZ the/DT tool/NN in/IN
the/DT picture/NN ?/."
    },
    "NE_tagging": {
      "NE_tagging_set": "CST's named entity recogniser,
https://cst.dk/online/navnegenkenderCSTNER/uk/",
      "NE_tagging_result": " [What,misc,uncertain] is the tool in the
picture ?"
    },
    "dependency_tagging": {
      "dependency_tagging_set": "CG-dependency,
https://edu.visl.dk/visl/en/parsing/automatic/dependency.php ",
      " dependency_tagging_result": "<ß>nWhat [what] <clb> <*>
<interr> INDP S/P @SC> #1->2nis [be] <mv> V PR 3S @FS-QUE #2->0nthe
[the] <def> ART S/P @>N #3->4ntool [tool] <tool> <def> <nhead> N S NOM
@<SUBJ #4->2nin [in] <advl-fs> PRP @<ADVL #5->2nthe [the] <def> ART
S/P @>N #6->7npicture [picture] <pict> <repr> <def> <nhead> N S NOM
@P< #7->5n? [?] PU @PU #8->0n</ß>"
    },
    "SRL_tagging": {
      "SRL_tagging_set": "HanLP,
https://hanlp.hankcs.com/en/demos/srl.html",
      "SRL_tagging_result": " What/ARG2 is/PRED (the tool in the
picture)/ARG1 ?"
    }
  }
}
```


Sentence 2: What is the nickname of the person in the picture?


```
{
  "meaning": {
```

```
    "POS_tagging": {
      "POS_tagging_set": "CST's Part-Of-Speech tagger (Brill, with
adaptations)",
      "POS_tagging_result": " What/WP is/VBZ the/DT nickname/NN of/IN
the/DT person/NN in/IN the/DT picture/NN ?/."
    },
    "NE_tagging": {
      "NE_tagging_set": "HanLP,
https://hanlp.hankcs.com/en/demos/ner.html ",
      "NE_tagging_result": ""
    },
    "dependency_tagging": {
      "dependency_tagging_set": "CG-dependency,
https://edu.visl.dk/visl/en/parsing/automatic/dependency.php ",
      " dependency_tagging_result": " <ß>nWhat [what] <clb> <*>
<interr> INDP S/P @SC> #1->2nis [be] <mv> V PR 3S @FS-QUE #2->0nthe
[the] <def> ART S/P @>N #3->4nnickname [nickname] <ac-cat> <def>
<nhead> N S NOM @<SUBJ #4->2nof [of] <np-close> PRP @N< #5->4nthe
[the] <def> ART S/P @>N #6->7nperson [person] <H> <def> <nhead> N S
NOM @P< #7->5nin [in] <advl-fs> PRP @<ADVL #8->2nthe [the] <def> ART
S/P @>N #9->10npicture [picture] <pict> <repr> <def> <nhead> N S NOM
@P< #10->8n? [?] PU @PU #11->0n</ß>"
    },
    "SRL_tagging": {
      "SRL_tagging_set": "HanLP,
https://hanlp.hankcs.com/en/demos/srl.html",
      "SRL_tagging_result": " What/ARG2 is/PRED (the nickname of the
person in the picture)/ARG1 ?"
    }
  }
}
```


Sentence 3: What is the job of the person on the left hand-side in the picture?


```
{
  "meaning": {
    "POS_tagging": {
      "POS_tagging_set": "CST's Part-Of-Speech tagger (Brill, with
adaptations)",
      "POS_tagging_result": " What/WP is/VBZ the/DT nickname/NN of/IN
the/DT person/NN in/IN the/DT picture/NN ?/."
    },
    "NE_tagging": {
      "NE_tagging_set": "HanLP,
https://hanlp.hankcs.com/en/demos/ner.html ",
      "NE_tagging_result": ""
    },
    "dependency_tagging": {
      "dependency_tagging_set": "CG-dependency,
https://edu.visl.dk/visl/en/parsing/automatic/dependency.php ",
      " dependency_tagging_result": " <ß>nWhat [what] <clb> <*>
<interr> INDP S/P @SC> #1->2nis [be] <mv> V PR 3S @FS-QUE #2->0nthe
[the] <def> ART S/P @>N #3->4nnickname [nickname] <ac-cat> <def>
<nhead> N S NOM @<SUBJ #4->2nof [of] <np-close> PRP @N< #5->4nthe
[the] <def> ART S/P @>N #6->7nperson [person] <H> <def> <nhead> N S
```

```
NOM @P< #7->5nin [in] <advl-fs> PRP @<ADVL #8->2nthe [the] <def> ART
S/P @>N #9->10npicture [picture] <pict> <repr> <def> <nhead> N S NOM
@P< #10->8n? [?] PU @PU #11->0n</ß>"
    },
    "SRL_tagging": {
      "SRL_tagging_set": "HanLP,
https://hanlp.hankcs.com/en/demos/srl.html",
      "SRL_tagging_result": " What/ARG2 is/PRED (the nickname of the
person in the picture)/ARG1 ?"
    }
  }
}
```


Sentence 4: What is the family name of the person in the centre of the picture?


```
{
  "meaning": {
    "POS_tagging": {
      "POS_tagging_set": "CST's Part-Of-Speech tagger (Brill, with
adaptations)",
      "POS_tagging_result": " What/WP is/VBZ the/DT family/NN name/NN
of/IN the/DT person/NN in/IN the/DT centre/NN of/IN the/DT
picture/NN ?/."
    },
    "NE_tagging": {
      "NE_tagging_set": "
https://cst.dk/online/navnegenkenderCSTNER/uk/",
      "NE_tagging_result": " [What,misc,uncertain] is the family name
of the person in the centre of the picture ?"
    },
    "dependency_tagging": {
      "dependency_tagging_set": "CG-dependency,
https://edu.visl.dk/visl/en/parsing/automatic/dependency.php ",
      " dependency_tagging_result": " <ß>nWhat [what] <clb> <*>
<interr> INDP S/P @SC> #1->2nis [be] <mv> V PR 3S @FS-QUE #2->0nthe
[the] <def> ART S/P @>N #3->5nfamily [family] <HH> <comp1> <comp1>
<ncomp> N S NOM @>N #4->5nname [name] <ac-cat> <comp2> <def> <nhead> N
S NOM @<SUBJ #5->2nof [of] <np-close> PRP @N< #6->5nthe [the] <def>
ART S/P @>N #7->8nperson [person] <H> <def> <nhead> N S NOM @P<
#8->6nin [in] <advl-fs> PRP @<ADVL #9->2nthe [the] <def> ART S/P @>N
#10->11ncentre [centre] <Labs> <inst> <def> <nhead> N S NOM @P<
#11->9nof [of] <np-close> PRP @N< #12->11nthe [the] <def> ART S/P @>N
#13->14npicture [picture] <pict> <repr> <def> <nhead> N S NOM @P<
#14->12n? [?] PU @PU #15->0n</ß>"
    },
    "SRL_tagging": {
      "SRL_tagging_set": "HanLP,
https://hanlp.hankcs.com/en/demos/srl.html",
      "SRL_tagging_result": " What/ARG2 is/PRED (the family name of
the person in the centre of the picture)/ARG1 ?"
    }
  }
}
```


Sentence 5: What is the name of the square in the picture?

```
{
  "meaning": {
    "POS_tagging": {
      "POS_tagging_set": "CST's Part-Of-Speech tagger (Brill, with
adaptations)",
      "POS_tagging_result": " What/WP is/VBZ the/DT name/NN of/IN
the/DT square/NN of/IN the/DT picture/NN ?/."
    },
    "NE_tagging": {
      "NE_tagging_set": "
https://cst.dk/online/navnegenkenderCSTNER/uk/",
      "NE_tagging_result": "[What,misc,uncertain] is the name of the
square of the picture ?"
    },
    "dependency_tagging": {
      "dependency_tagging_set": "CG-dependency,
https://edu.visl.dk/visl/en/parsing/automatic/dependency.php ",
      " dependency_tagging_result": " n<ß>nWhat [what] <clb> <*>
<interr> INDP S/P @SC> #1->2nis [be] <mv> V PR 3S @FS-QUE #2->0nthe
[the] <def> ART S/P @>N #3->4nname [name] <ac-cat> <sem-c> <def>
<nhead> N S NOM @<SUBJ #4->2nof [of] <np-close> PRP @N< #5->4nthe
[the] <def> ART S/P @>N #6->7nsquare [square] <Lh> <geom> <def>
<nhead> N S NOM @P< #7->5nof [in] <np-close> PRP @N< #8->7nthe [the]
<def> ART S/P @>N #9->10npicture [picture] <pict> <repr> <def> <nhead>
N S NOM @P< #10->8n? [?] PU @PU #11->0n</ß>"
    },
    "SRL_tagging": {
      "SRL_tagging_set": "HanLP,
https://hanlp.hankcs.com/en/demos/srl.html",
      "SRL_tagging_result": " What/ARG2 is/PRED (the name of the
square of the picture)/ARG1 ?"
    }
  }
}
```


9.10 Intention JSON Files

Q1: What is the tool in the picture?


```
{
  "Intention":{
    "qtopic": "tool",
    "qfocus":"What",
    "qLAT":"tool",
    "qSAT":"ETC",
    qdomain":"everyday life"
  }
}
```


Q2: What is the nickname of the person in the picture?


```
{
  "Intention": {
    "qtopic": "person",
```

```
    "qfocus":"What",
    "qLAT":"nickname",
    "qSAT":"PS_NAME",
    "qdomain":"famous people"
  }
}
```


Q3: What is the job of the person on the left hand-side in the picture


```
{
  "Intention":{
    "qtopic":"person",
    "qfocus":"What",
    "qLAT":"job",
    "qSAT":"CV_OCCUPATION",
    "qdomain":"famous people"
  }
}
```


Q4: What is the family name of the person in the centre of the picture?


```
{
  "Intention":{
    "qtopic":"person",
    "qfocus":"What",
    "qLAT":"family name",
    "qSAT":"PS_NAME",
    "qdomain":"famous people"
  }
}
```


Q5: What is the name of the square in the picture?


```
{
  "Intention":{
    "qtopic":"square",
    "qfocus":"What",
    "qLAT":"square",
    "qSAT":"LC_TOUR",
    "qdomain":"traveling"
  }
}
```