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

|  |  |
| --- | --- |
|  | **Public document** |
| **N704** | 2022/05/18 |
| **Source** | Requirements (MCS) |
| **Title** | XR Theatre Use Case – Model and components |
| **Target** | MPAI Community |

# A Real-to-Virtual Interaction Model

The Model depicted in Figure 1 assume a complete symmetry between the actions performed and the data exchanged between the Real World and a Virtual World.

|  |  |
| --- | --- |
| The Real to the Virtual World workflow is:   1. Signals from the Real World are captured and digitally in an appropriate format. 2. Data are analysed and features extracted. 3. Features are interpreted. 4. Actions are generated. 5. Virtual Experiences are generated. 6. and converted to aformat suitable for the Virtual World. | The Virtual to Real-World workflow is:   1. Data from the Real World are captured and converted to an appropriate format. 2. Data are analysed and features extracted. 3. Features are interpreted. 4. Actions are generated. 5. Real Experiences are generated. 6. and converted to a format suitable for the Real World. |

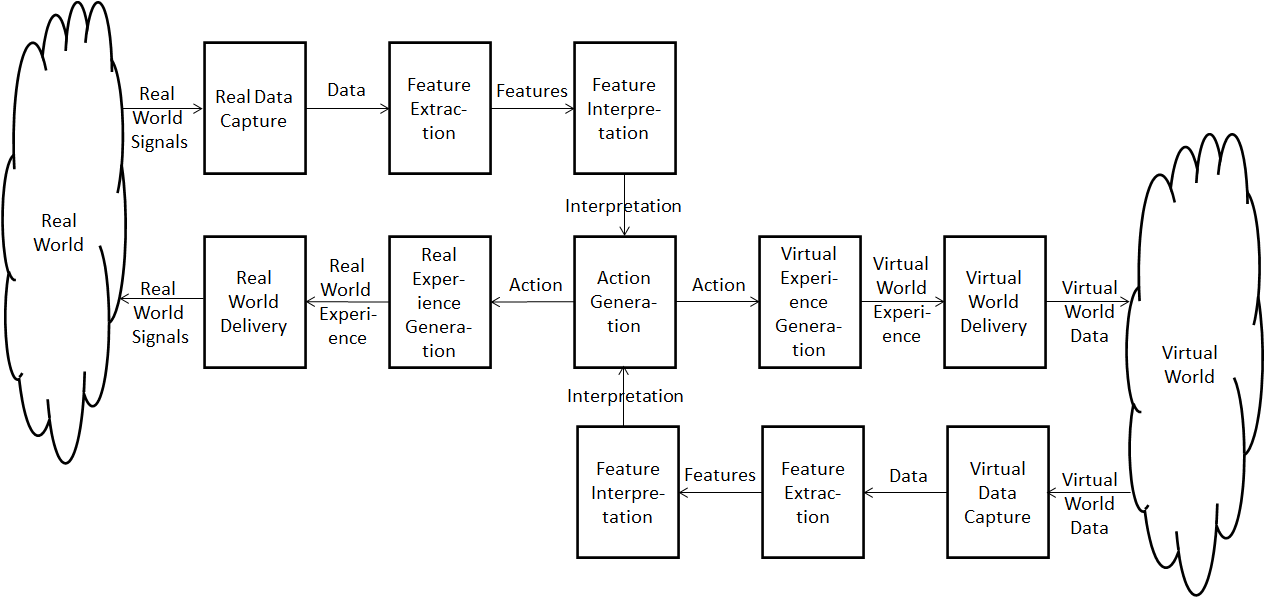


Figure 1 – General Reference Model of the Real-to-Virtual-to-Real Interaction

The functions of the blocks are described in the following table.

|  |  |
| --- | --- |
| **Block name** | **Function** |
| Real Data Capture | Converts information from the Real World into digital information in an appropriate format. |
| Feature Extraction | Extracts features from the Data acquired. |
| Features Interpretation | Interprets the features. |
| Action Generation | Generates an action based on the interpreted features |
| Virtual Experience Generation | Generates a data set the corresponds to an experience implemented in the Virtual World. |
| Virtual World Delivery | Converts the Virtual Experience into a format suitable to the Virtual World. |

The path from the Virtual to the Real World is entirely simmetric with the Virtual to the Real World.

In the XR Theatre Use Case, it is necessary to introduce a new function called

|  |  |
| --- | --- |
| Script | Contains a collection of descriptors that the director/producer selects for execution at runtime as a sequence of descriptors that control the action/experience in both the Real and Virtual Worlds. |

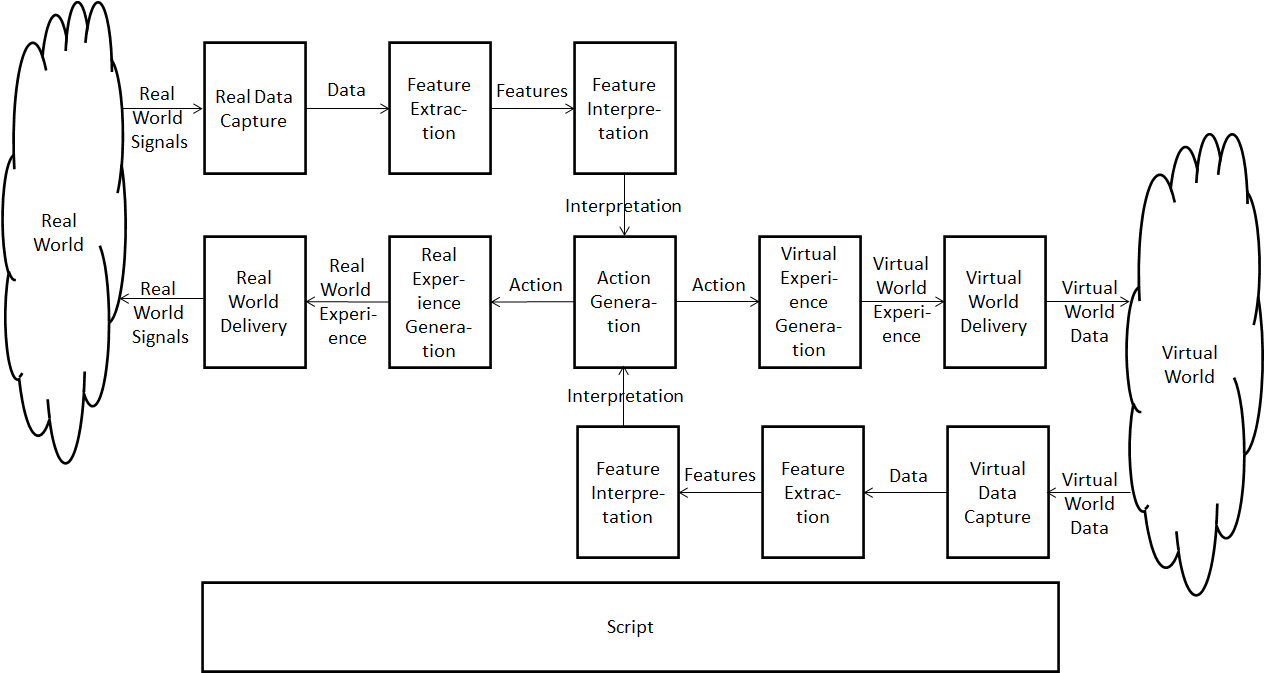


Figure 2 – Reference Model of the XR Theatre Use Case

# Devices and data in the XR Theatre Use Case

Figure 5 gives a comprehensive view of the devices and data relevant to the XR Theatre Use Case.

Diagram

Description automatically generated

Figure 5 - Devices and data of the XR Theatre Use Case

# Major use case

VJ/DJ/Console operator (real-time show control team) inputs command into systems via:

1. Buttons
2. Joysticks
3. Controllers (inertial, proximity, consoles etc.)
4. Hand gestures
5. Headset

The commands are input into the Action Generator Engine (AGE) to modify the multisensory elements (visual, audio, …) of both the Real and the Virtual World Experiences, including:

1. Video/audio clips from media servers.
2. Real-time 2D/3D generated graphics/audio (from a game engine, AI engines generating 3D).
3. Additional video/audio sources
4. 3D objects, characters, scenes.
5. Lighting and special effects.
6. Other experiential elements in both the Real and Virtual Worlds.

Multisensory elements may further be modified by additional real-world and virtual-world data, including:

1. Audience behaviour.
2. Performer behaviour.
3. Events on the stage/dome/metaverse.

Possible sub-use cases (Real/Virtual versions possible):

1. DJ/VJ performance at a dance party.
2. Live theatrical stage performance.
3. Live concert performance.
4. Immersive art experience.
5. Sporting events (e.g., esports).
6. Experiential marketing/branding.
7. Meetings/presentations.
8. Experiential retail/shopping.

## Esports tournament

**Purpose**

To define interfaces between components in order to enable an XR Theatre (RW) to host any pre-existing VW game for the purpose of hosting an esports tournament with RW and VW audience interactivity. To the extent that the game possesses the required interfaces, the XR Theatre can drive action within the VW.

**Description**

Two teams of 5 RW players are arranged on either side of a RW stage, each using a computer to compete within a common real-time Massively Multiplayer Online (MMO) VW game space where the 10 players are represented by avatars each driven by a player with a role (e.g., magicians, warriors, soldier, etc.), properties (e.g., costumes, physical form, physical features) and actions (e.g., casting spells, shooting, flying, jumping) operating in the VW which is populated by environmental structures (e.g., terrain, mountains, bodies of water), and autonomous characters (e.g., dragon, monsters, various creatures). Multiple VW cameras follow the action which is then projected onto an immersive screen surrounding RW spectators and live-streamed to remote spectators who experience the 2D videos and all related sounds of the VW game space.

A shoutcaster calls the action as the game proceeds. The image of RW players, player stats or other information or imagery may also be displayed on the immersive screen and the live stream. Additionally, the RW tournament space is augmented with lighting and special effects, music and costumed performers.

Live stream viewers interact with one another and with commentators through live chats, Q&A sessions, etc. RW spectators also interact through shouting, waving and interactive devices (e.g., LED wands, smartphones) which can be

1. captured via camera/microphone or wireless data interface (see RW data in Figure 2) providing data.
2. data are processed to extract features
3. features are interpreted
4. RW/VW actions can be generated as a result of
   1. in-person or remote audience behaviour (RW), or
   2. data collected from VW action (e.g., spell casting, characters dying, bombs exploding)

At the end of the tournament, there is an award ceremony featuring the winning players on the RW stage with great fanfare.

**Role of AI**

1. Controlling the PoV of spectator cameras based on actions and environmental structures within the VW, including player behaviour, autonomous characters, the environment, the importance of the action, spectator response, and event audio.
2. Controlling game environment and autonomous characters.

# Earlier notes

Table 1 analyses the data generated by the left-hand side of Figure 5 using the 4 stages of data processing identified in Figure 3.

Data is generated by humans in different instances:

1. Video Jockey
2. Performer
3. Audience

Table 1 - Data from the Real World

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data types** | **Raw data** | **Features** | **Interpretation** | **Action** |
| Audio | Mono, stereo, multichannel | Audio scene description |  |  |
|  |  | Frequency analysis |  |  |
| Speech |  | Speech scene | Text (ASCII) | Translation |
|  |  |  | Emotion |  |
|  |  |  | Intent |  |
|  |  |  | Meaning |  |
| Video | Generic video scene | Video scene description |  |  |
|  |  | Objects or body | Bodyàs skeleton |  |
|  |  | Objects’ position-orientation & derivatives |  |  |
|  | Entire body | Objects’ position-orientation & derivatives |  | Avatar Animation |
|  | Torso |  |  |  |
|  | Gesture | Keypoints | Sign language | Avatar Animation |
|  | Face |  | Identification |  |
|  |  | FACS | Expression | Avatar Animation |
|  | Eye | Eye tracking | Viewer’s interest |  |
| Body sensors | Raw data from sensors (positional, 3DoF-6DoF) | Relative position and orientation of body parts - Skeleton | Interpreted gestures  What the skeleton does expressed in a (choreography) language | Avatar Animation |
| Biometric | EEG waveforms |  | Thought |  |
|  | Heart rate over a period of time beats/min | Heart rate variability | Excitation, anger, pleasure |  |
| Manual control | Button | on-off | (context-dependent) |  |
|  | Joystick | x-y-z values | (context-dependent) |  |
|  | Camera control | PoV x’-y’-z’ values  FoV angle, Interocular separation and orientation cm and angle | Interpretation of the operator’s intention when capturing the scene |  |
|  | Controller |  |  |  |

Table 2 analyses the data going to the left-hand side of Figure 5.

Data consumed by humans come from the Metaverse or from the Experience Generator.

Table 2 - Data to the Real World

|  |  |
| --- | --- |
| **Data types** |  |
| Audio | Natural or synthetic sound |
| Speech | Natural or synthetic speech |
| Full MV visual experience |  |
| Haptics | Force feedback |
|  | touch |
|  | vibration |
|  | haptic |
|  | vest |
|  | glove |
| Vestibular feedback | 6DoF motion platform |

Scope of the use case

1. Capture, processing, interpretation and action of actors (audience, performer and VJ) data:
   1. Audience interactivity
      1. Capturing the “sentiment”, “behaviour”, “mood”, “intent” (in the following called “sentiment etc.”) of an audience (common or individual, average, variance) including behaviour (gestures, facial), biometric (e.g., heart rate, brainwaves), audiometer (cheering, booing), buttons and controllers at the seats. It is called “Reading the room”. One solution is supervised learning of a machine by feeding it with many instances of labelled “common” sentiments. There is also the possibility of unsupervised learning. If we do it, we would need to define the type of signals that we use to measure the “sentiment, etc.” to train the machine and the human-recognisable categories of “sentiment etc.” and exactly what “sentiment etc.” means. “Sentiment etc.” is dependent on culture. Compared to capturing the “sentiment etc.” of an individual done in MPAI-MMC, here we are measuring the sentiment of a crowd.
      2. Measuring the “response”, “affect”, caused by a known stimulus. This is a system-level problem. We present something that influences a crowd and may influence what is being presented. The result may be adapting the multisensory (e.g., visual audio, aroma, haptic, vibration, fog, special effects, performer’s movement) stimulus based on the sentiment etc. captured.
   2. Performer’s interactivity.
      1. Gesture recognition and interpretation of the “meaning” of a performer’s movement, e.g., sign language, dance notation, mudras language are examples of how a stimulus can be captured and interpreted. A machine could be trained with labelled performer movement for the required instances of cultures and contexts.
      2. The trained machine would provide the interpreted gestures. Another machine can be trained to adapt the multisensory stimulus based on the interpreted gesture.
   3. VJ interactivity
      1. Gesture capture, recognition and interpretation of the “meaning” of VJ movements and, gestures. A machine could be trained with labelled performer movement for the required instances of cultures and contexts.
      2. Another machine receives the interpreted gestures and is trained to adapt the multisensory stimulus based on the interpreted gesture.
2. Raw data interface to/from the Metaverse. The MV exists with its own platform and its two-way interface (viewing, controlling and receiving a reaction from the MV). User’s multisensory experience (input and output) can include:
   1. Audio, may be the result of pre-processing to get the intended audio
      1. in: speech, my music, spatial audio field;
      2. out: sound from the MV
   2. Visual, may be the result of pre-processing to get the intended visual info
      1. in: my face, my entire body;
      2. out: the entire visual experience of a MV room from the PoV of the avatar; interocular separation, foveation
   3. Gesture
      1. in: kinetic tracking of physical body parts;
      2. out: force feedback from a robot, a subset of haptics
   4. Position, acceleration
      1. in: coordinates+derivative, orientation+derivative in the user’s space
   5. Vestibular feedback
      1. out: 6DoF motion platform
   6. Haptic (out: touch, vibration, haptic vest, glove)
   7. Manual controls (in: e.g., buttons, joysticks)
   8. Biometric (in: e.g., eye tracking, EEG, EOG)
   9. Headset (in/out: some of the above features in an integrated fashion)
   10. Spectator camera control. A “camera operator” is provided specialised control to select and animate (e.g., by walking in the metaverse) the PoV of a virtual scene to broadcasts the corresponding scene.
       1. in: e.g., camera control parameters, position, orientation, field of view.
       2. out: audio-visual.
3. Processing of the raw data to act on the metaverse in a variety of ways:
   1. Interpretation of the data by extracting information such as
      1. Emotion, e.g., speech intonation, facial expression
      2. Non-verbal communication, face (nod, wink etc), sign language
      3. Intent
      4. Meaning
   2. Action following interpretation
      1. Manipulation of avatars
      2. Modification of virtual environment
4. Multichannel audio environment management. Multiple audio sources from geographically separated places send their audio through the internet. How to combine the sound signals in such a way that synchronisation with universal time is preserved with an accuracy of 10 ms (to achieve musicality as opposed to a real-time conversation between two remote people).

# The Real-to-Virtual line

Figure 3 depicts the Real-to-Virtual processing line of Figure 1.

Graphical user interface, application, Word

Description automatically generated

Figure 3 - The Real-to-Virtual line

The scope of the 4 steps is not uniquely and universally defined. There should be no doubt that PCM Video is the output of Data Capture, but is video compression part of Data Capture? Possibly, DPCM-coded video is, but sophisticated motion-compensation video compression should probably be part of Feature Extraction.

Examples are:

1. In a human-machine conversation, the machine has asked “should I go?”. The machine captures the video of the head of the human and interprets that the human is nodding. The machine realises that nodding (e.g., in conjunction with a speech-based dialogue) means that it should go and executes.

*Human face – Face feature extraction – Face feature interpretation – Action (Go)*

1. A human controls the cursor on a screen wearing an EEG cap. The machine extracts EEG features, then interprets the EEG features and understands that the human is thinking of moving the cursor up. The machine executes.

*Signals from EEG – EEG feature extraction – EEG feature interpretation – Action (cursor up)*

# Borders between the Real and Virtual World

Raw data, i.e., the data captured by sensors, can enter a Virtual World directly or after processing steps are performed, i.e., after feature extraction or after feature interpretation or after action. Therefore. there are different configurations depending on where a particular function is, in the machine or in the metaverse:

1. The machine is an integral part of the metaverse receiving sensor data directly.
2. The machine is partly outside of the metaverse effecting only certain one of the following functions on raw data:
   1. Hands over features to the metaverse.
   2. Hands over interpreted features to the metaverse.
   3. Hands over action to the metaverse.

In the two examples above, if the machine is fully part of the metaverse, it receives the video of the human face or the EEG signals. Otherwise, it extracts the features from the raw data leaving their interpretation to the metaverse, or it interprets features leaving the conversion of features into action to the metaverse, or it converts the interpreted features into action and passes it to the metaverse.

This is depicted in Figure 4.

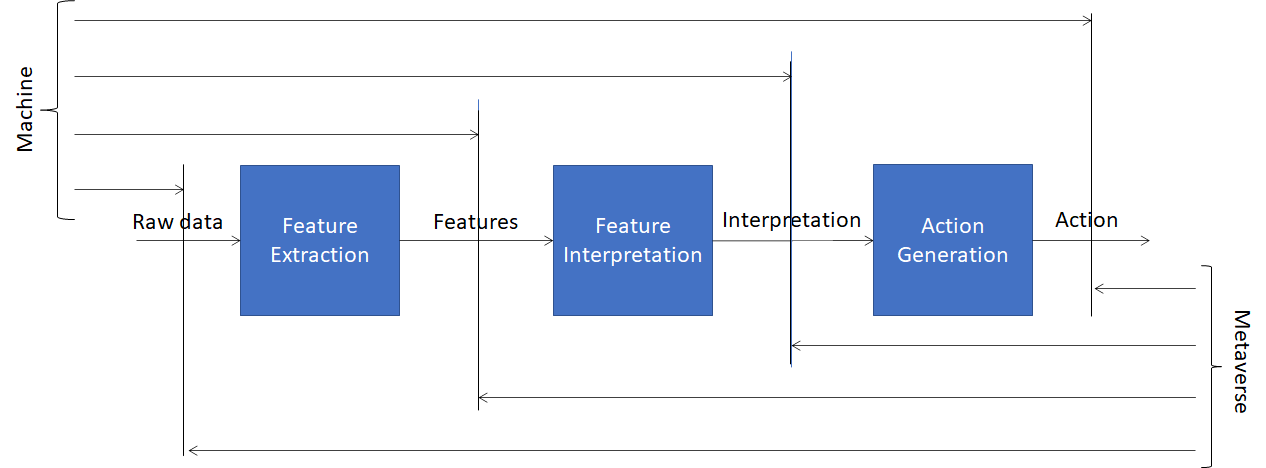


Figure 4 - The 3 data processing stages in the Real-to-Virtual line

Note that *Interpretation* can be culture-dependent and *Action* can be application-dependent.