Information Model for LAE in MAR (ISO/IEC NP 23490)

ISO/IEC JTC1 SC24 WG9 Meeting
20-25 January, 2019
Kwan-Hee Yoo
Chungbuk National University

History

ISO/IEC 23490 Information technology — Computer graphics, image processing and environmental representation — Information model of live actor and entity for mixed and augmented reality

Status	**Version Date**	Life cycle
NP Ballot	2018-07-10	
New Project	2019-01-17	20.00

Timeline

REGISTRATION DATE	TIMEFRAME	TIME SINCE REGISTRATION	IN STAGE
2019-01-17	24 months	0 day	20.00 for 0 day

Stage

Stage	Version	Description	Target date	Limit date	Started	Status
20.00	**1**	**New project registered in TC/SC work programme**			**2019-01-17**	**CURRENT**
30.00		Committee draft (CD) registered	2019-05-01			AWAITING
40.00		DIS registered	2019-12-18	2020-01-17		AWAITING
60.60		International Standard published	2020-12-22	2021-01-17		AWAITING

Show all stages

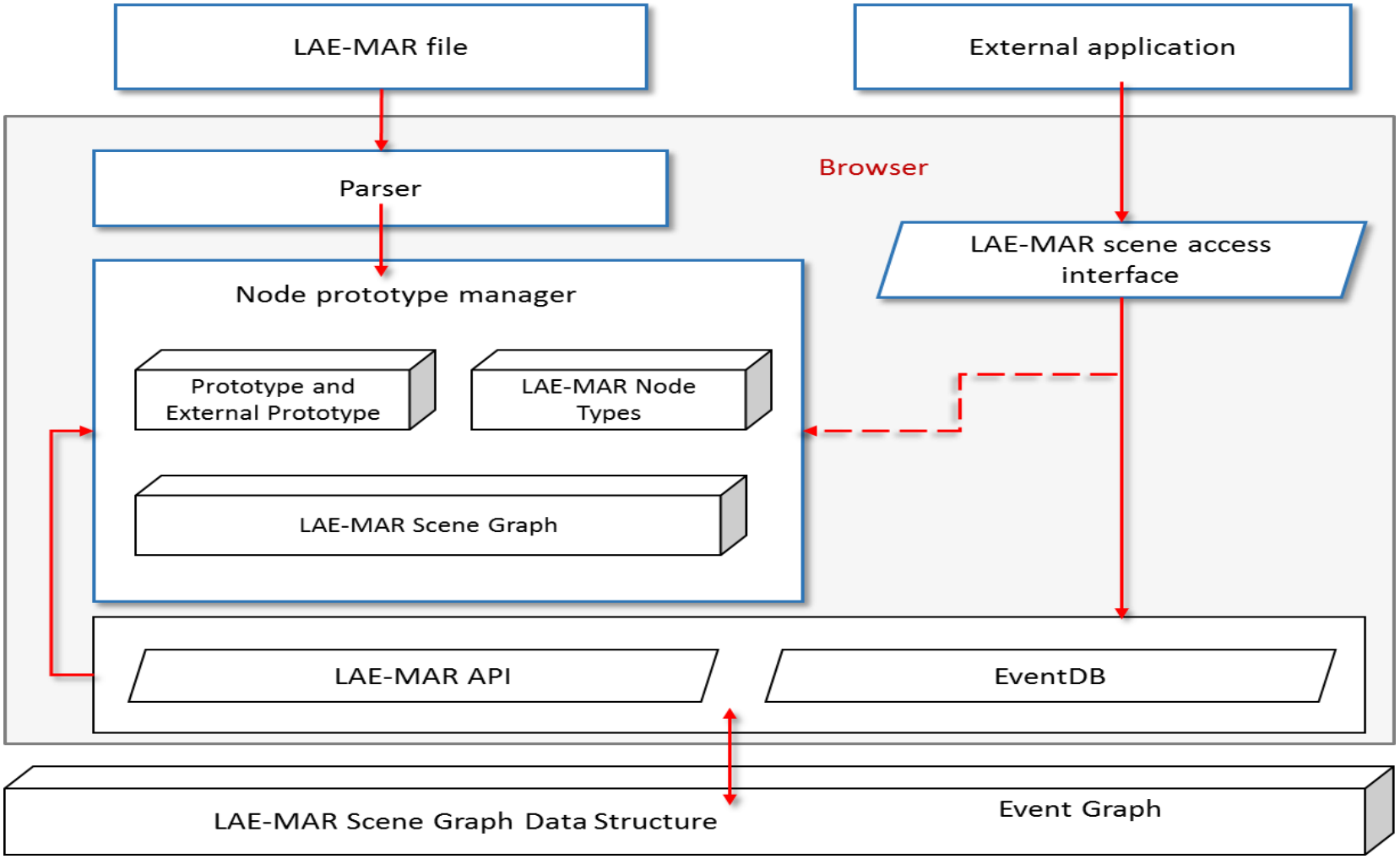
Introduction

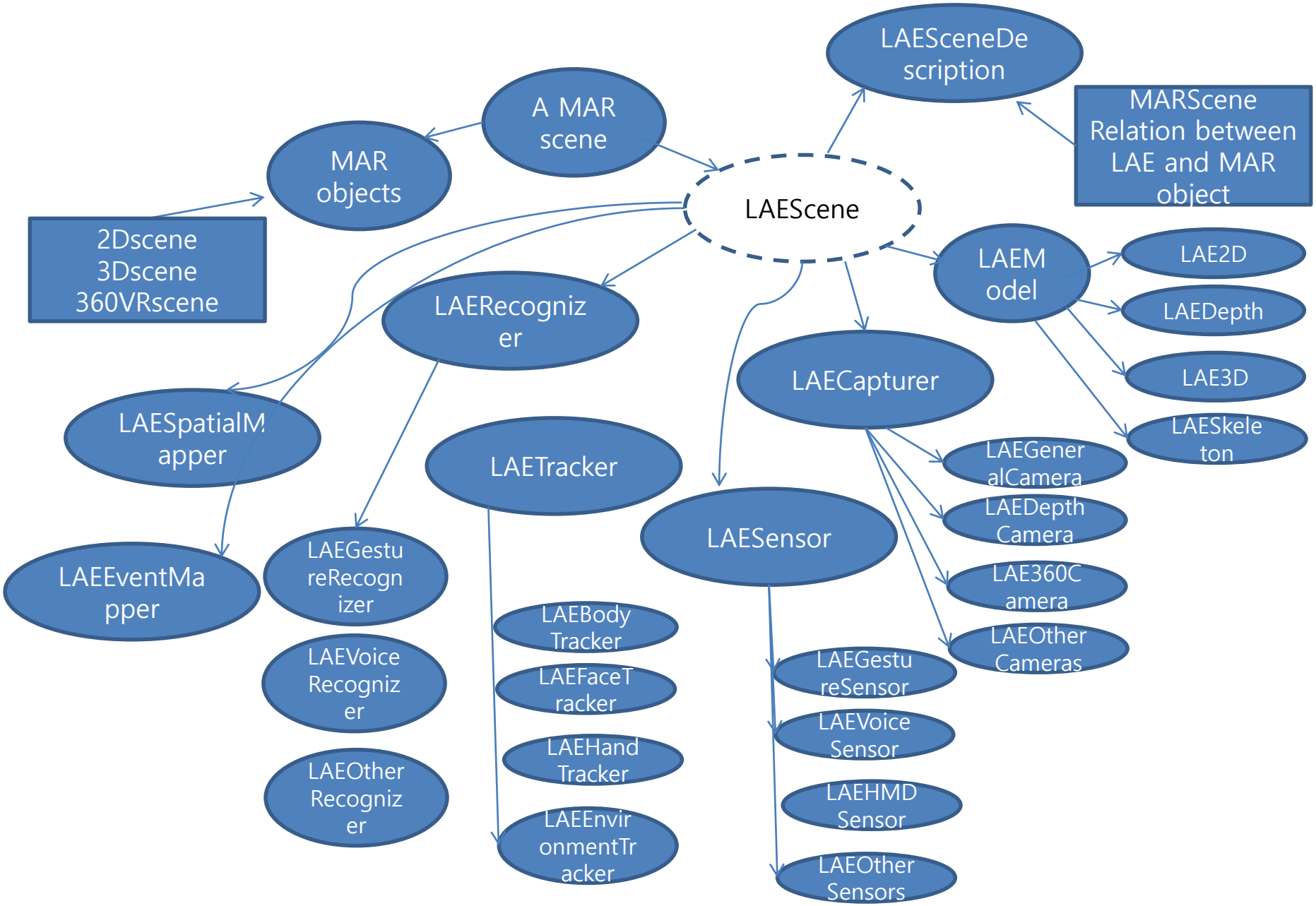
- illustrate the system architecture of LAE-MAR
- define the characteristic and model of LAE
- design LAE-MAR scene graph for supporting functionalities of each components
- define node definitions for LAE information model
- allow the specification to be implemented at varying levels of service
- provide alternative application programming interface (API)

Scope

- Mix and matching for expressional richness
- Compatibility and extendibility to existing constructs for VR and other mark-up documents
- Standardization for a file format of LAE contents in a LAE-MAR system
- Standardization for streaming or transmitting LAE contents among LAE-MAR systems.

LAE-MAR architecture




LAESceneDescription
A MAR scene
MAR objects
MARScene Relation between LAE and MAR object
2Dscene
3Dscene
360VRscene
LAEScene
LAEModel
LAE2D
LAEDepth
LAE3D
LAESkeleton
LAERecognizer
LAECapturer
LAESpatialMapper
LAETracker
LAESensor
LAEGeneralCamera
LAEDepthCamera
LAE360Camera
LAEOtherCameras
LAEEventMapper
LAEGestureRecognizer
LAEVoiceRecognizer
LAEOtherRecognizer
LAEBodyTracker
LAEFaceTracker
LAEHandTracker
LAEEnvironmentTracker
LAEGestureSensor
LAEVoiceSensor
LAEHMDSensor
LAEOtherSensors

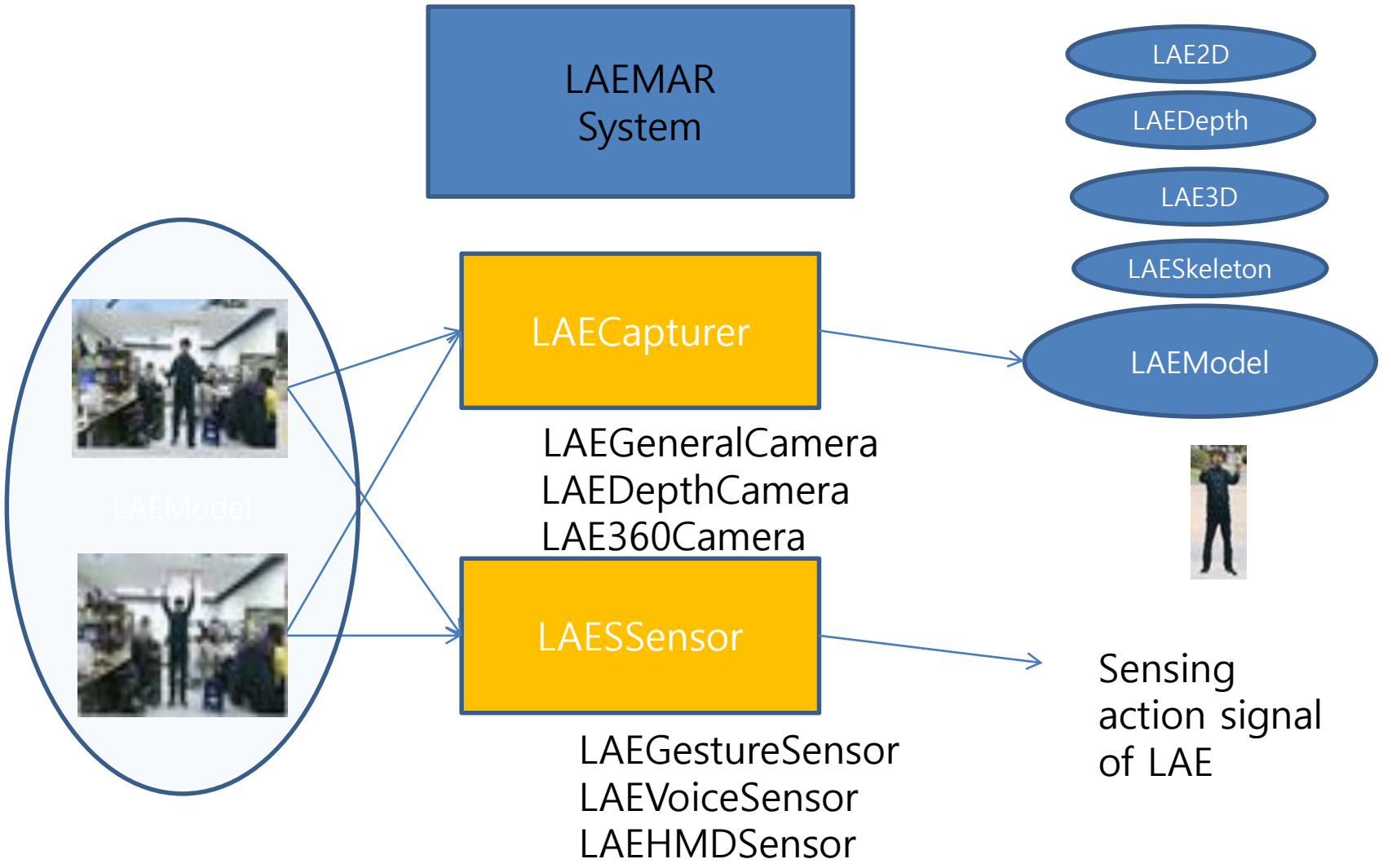
MR Game LAE is Participating

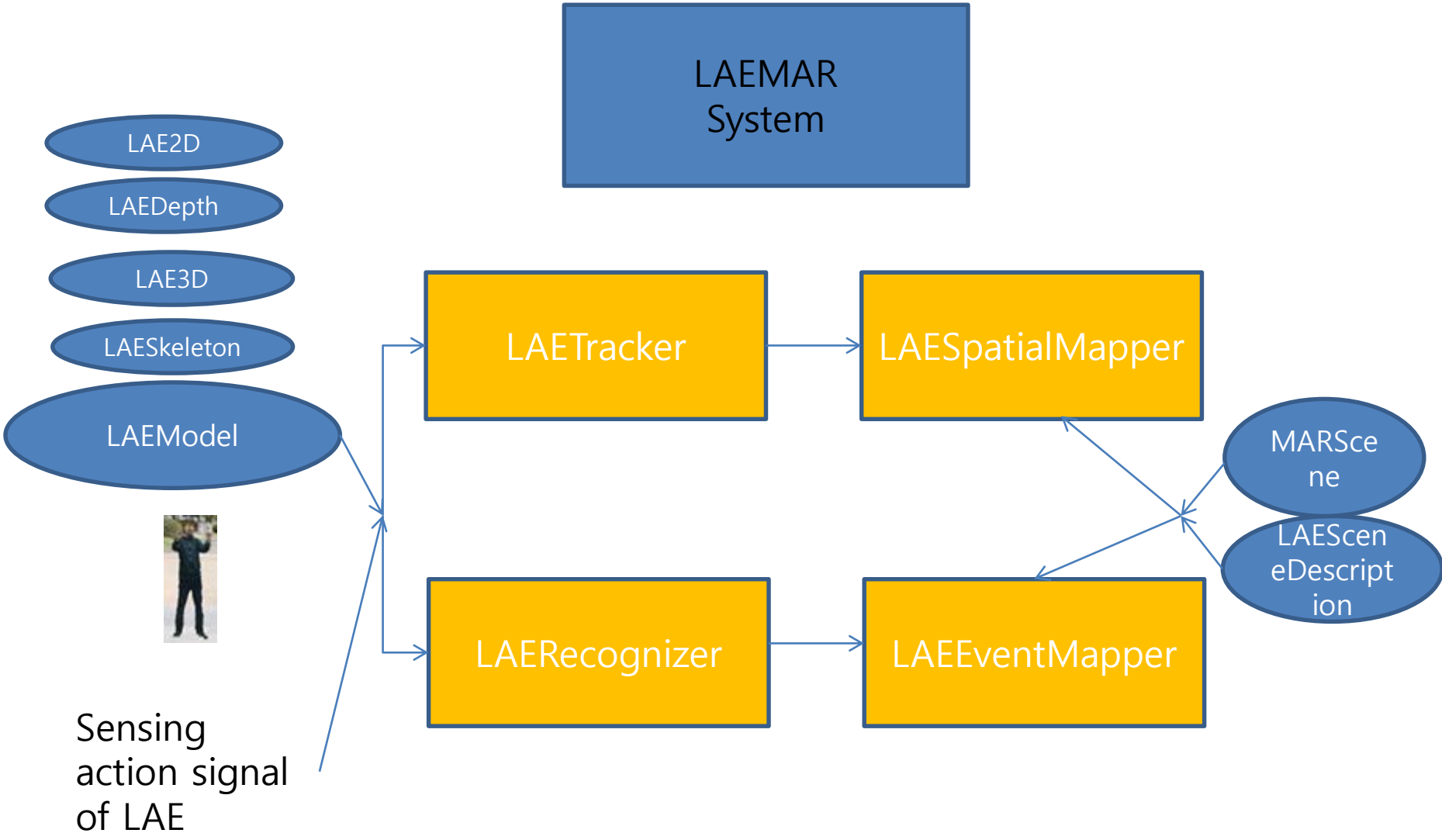
https://www.youtube.com/watch?v=Kw5yPyv9O9E

LAEScene::MARScene

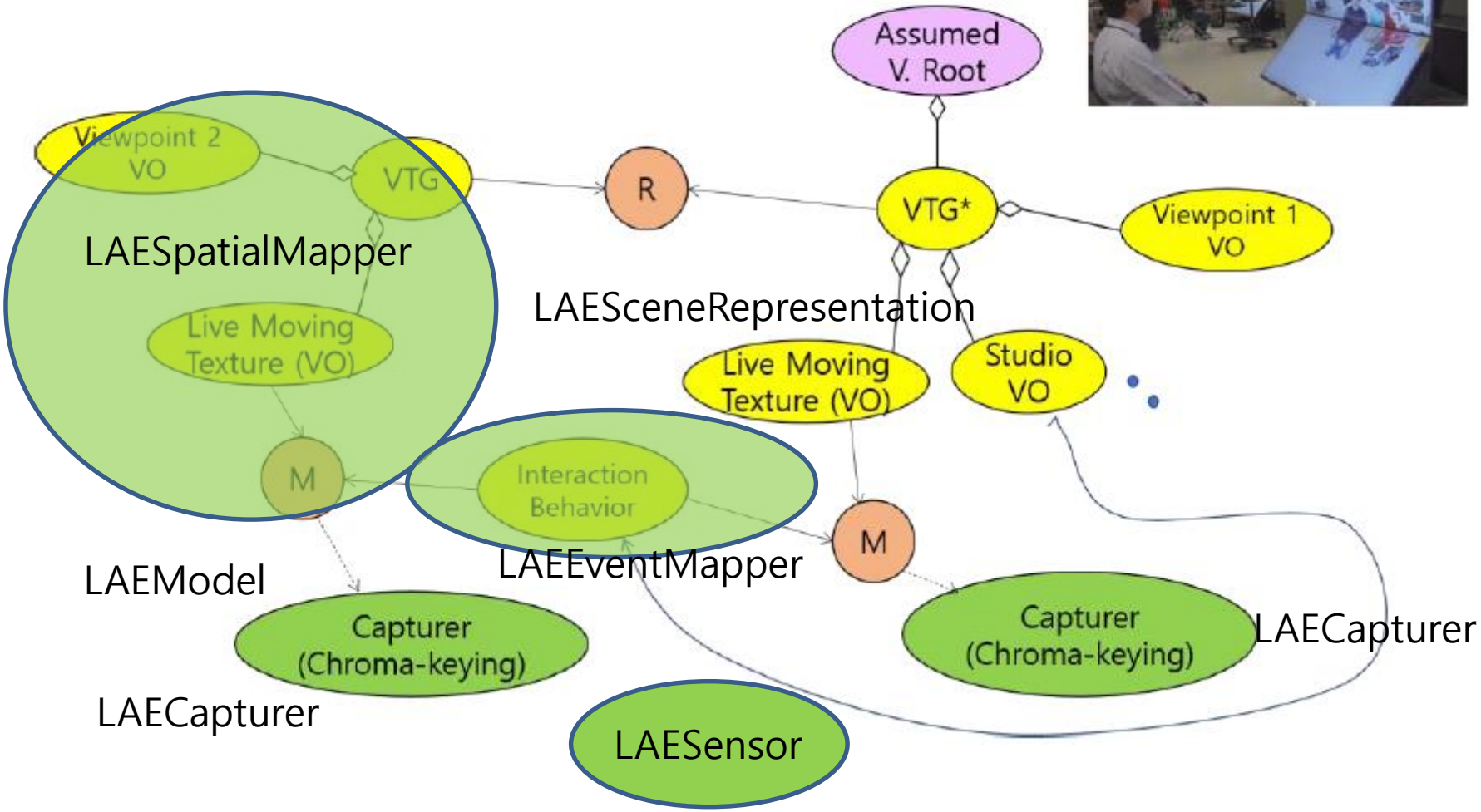
LAE-MAR Scene Graph

- An MAR scene
--------- MAR Object
---------- 3D Object
---------- Shape
---------- Material
---------- Geometry
--------- Display
--------- Renderer
--------- User Interface
---------- LAEModel
---------- LAE2D
---------- LAE3D
---------- LAEDepth
---------- LAESkeleton
---------- LAECapturer : MARCapturer
---------- LAECamera
---------- LAEGeneralCamera
---------- LAEDepthCamera
---------- LAE360Camera
---------- LAESensor: MARSensor
---------- LAEHMDSensor
---------- LAEPCHMD
---------- LAEMobileHMD
---------- LAECotrollerSensor
---------- LAETracker: MARTracker
---------- LAEBodyTracker
---------- LAEFaceTracker
---------- LAEHandTracker
---------- LAEEnvironmentTracker
---------- LAERecognizer: MARRcognizer
---------- LAESpatialMapper: MARSpatialMapper
---------- LAEEventMapper: : MAREventMapper
---------- LAESceneRepresentation: MARSceneRepresentation
---------- MARScene
---------- LAESpatialMappingInform (spatial relation between LAE and MAR scene)
---------- LAEEventMappingInform (event relation between LAE and MAR scene)


LAEMAR
System
LAE2D
LAEDepth
LAE3D
LAESkeleton
LAEModel
LAECapturer
LAEGeneralCamera
LAEDepthCamera
LAE360Camera
LAEModel
LAESSensor
LAEGestureSensor
LAEVoiceSensor
LAEHMDSensor
Sensing
action signal
of LAE


LAEMAR System
LAE2D
LAEDepth
LAE3D
LAESkeleton
LAEModel
LAETracker
LAESpatialMapper
MARScene
LAESceneDescription
LAERecognizer
LAEEventMapper
Sensing action signal of LAE

LAEMAR System
LAEScene::MARScene



LAE2D representation without chromakeying

Elements Console Sources Network Timeline Profiles Application Security Audits

```
<!DOCTYPE html>
<html lang="en">
▶<head>…</head>
▼<body style="margin: 0px; overflow: hidden;">
  ▼<lae id="lae1" description="LAE project" type="LAE2D" visible="true" showfps="true">
    <!-- LAE Display -->
    ▼<laeprojectiondisplay>
      <laeperspectiveprojection id="proj1" description="Projection Camera" position="0 150 400" fov="45" neardistance="0.1" fardistance="20000">
              </laeperspectiveprojection>
    </laeprojectiondisplay>
    <!-- LAE SceneRepresentation -->
    ▼<laescenerepresentation id="scene1" description="scene representation" scenetype="LAE360VRScene">
      ▼<lae360vrscene id="vrscene1" type="equirectangular">
        <equirectangularvr id="equir1" src="texture/u20170704.png" scale="-1 1 1" size="500 60 40"></equirectangularvr>
      </lae360vrscene>
    </laescenerepresentation>
    <!-- LAE2D -->
    <lae2d id="lae2D" laex="0" laey="0" visible="true"></lae2d>
    <!-- LAE Camera -->
    ▼<laecamera id="camera1" description="used for capturing LAE in real world" cameratype="depth">
      <laedepthcamera id="depthcam1" description="depth camera" camerasdk="kinectv2"></laedepthcamera>
    </laecamera>
    <!-- LAE Tracker -->
    ▼<laetracker id="tracker1" description="tracking module" trackingtype="body">
      <laebodytracker id="bodytrack" type="color"></laebodytracker>
    </laetracker>
    <!-- LAE SpatialMapper -->
    <laespatialmapper id="spatial1" position="0 -25 0"></laespatialmapper>
  </lae>
```

Figure 4.4 LAE2D representation without chromakeying

LAE2D representation without chromakeying

(a)

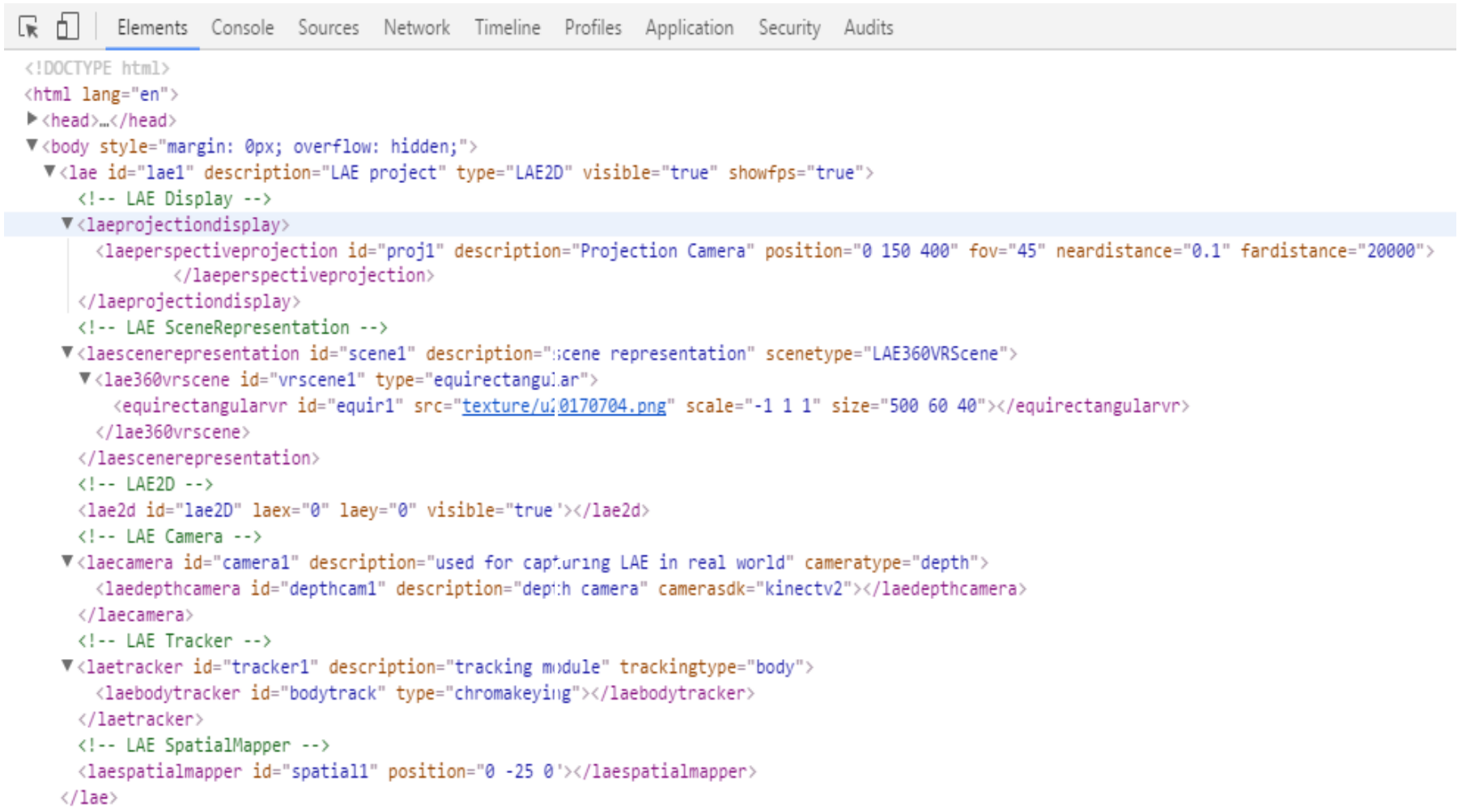
(b)

(c)

(d)

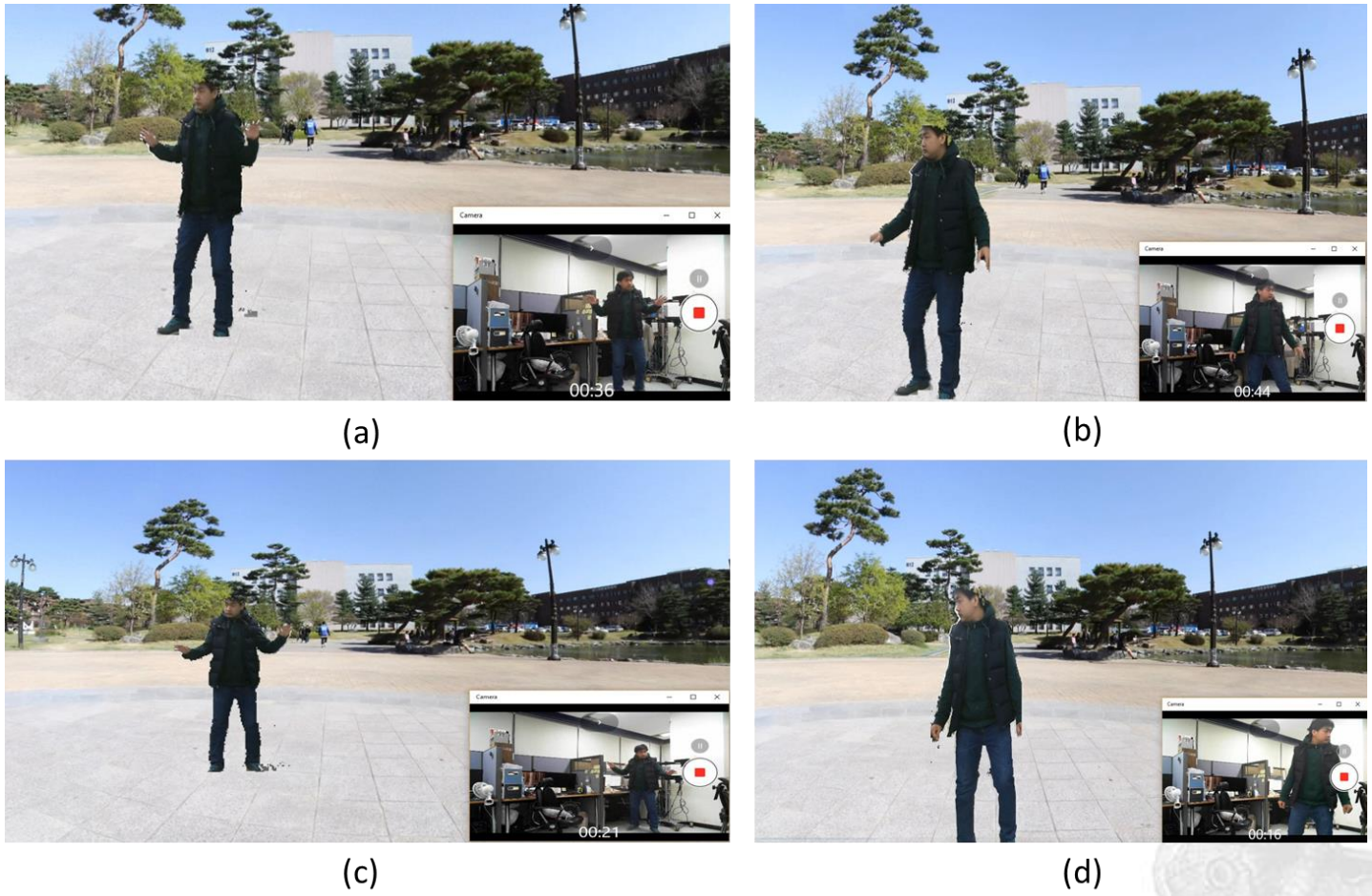
LAE2D representation without chromakeying

LAE2D movement in 360° virtual reality scene



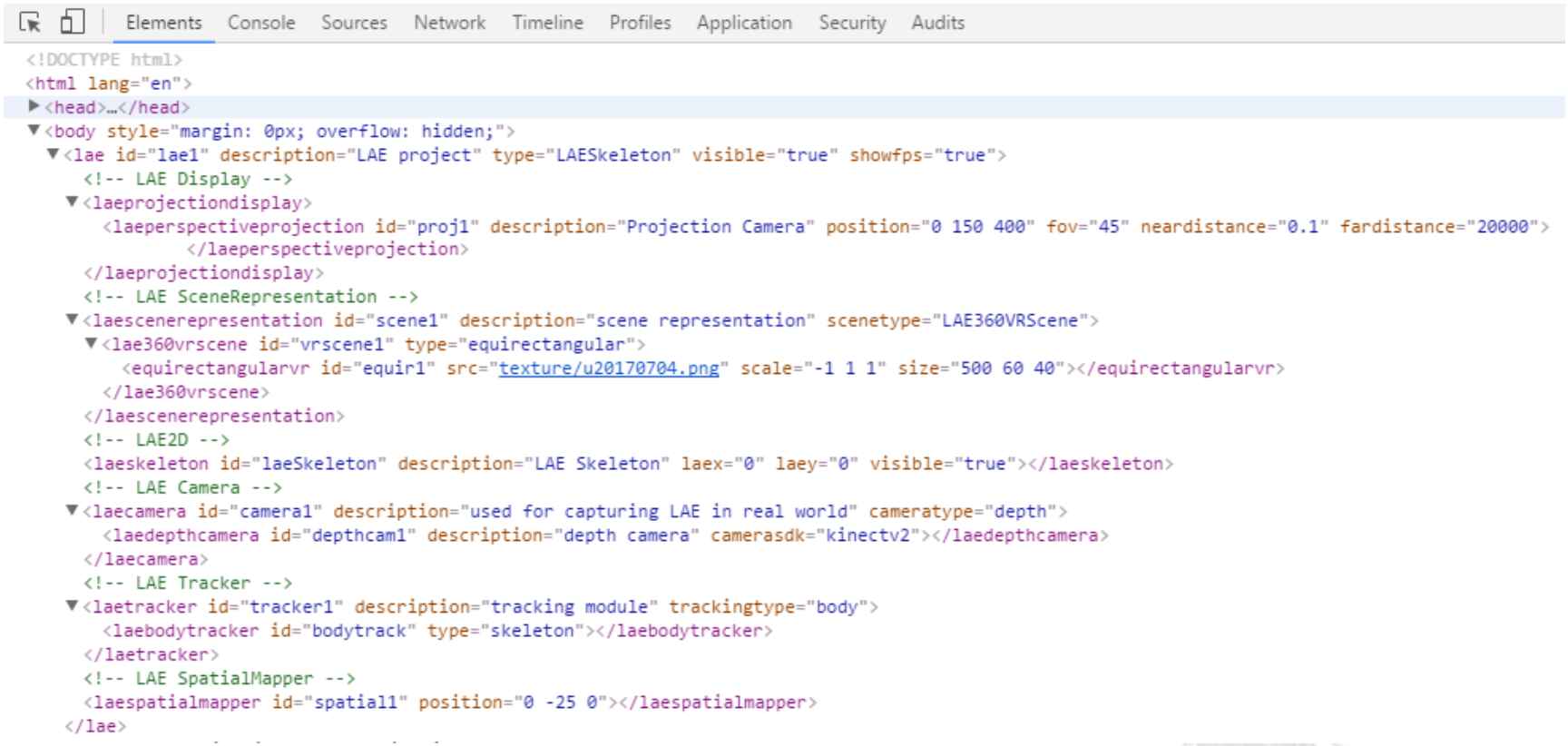
LAE movement in 360° virtual reality scene

LAE2D movement in 360° virtual reality scene



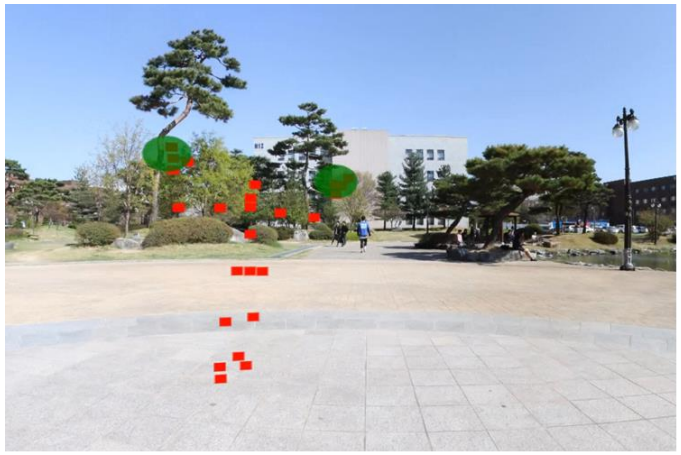
LAE movement in 360° virtual reality scene

LAESkeleton representation

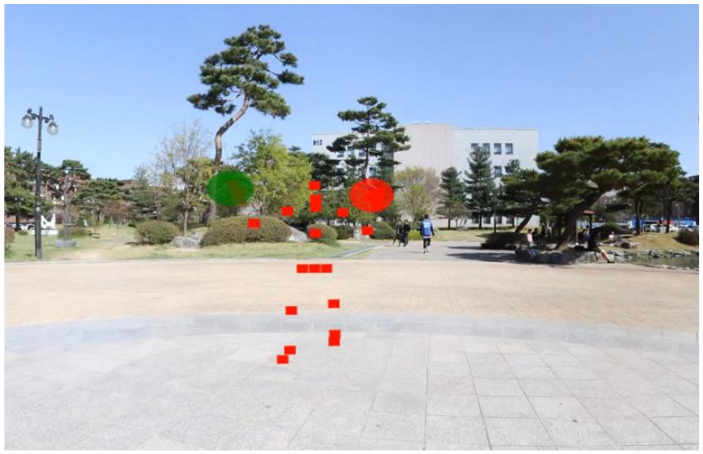


LAESkeleton representation and movement in 360º virtual reality

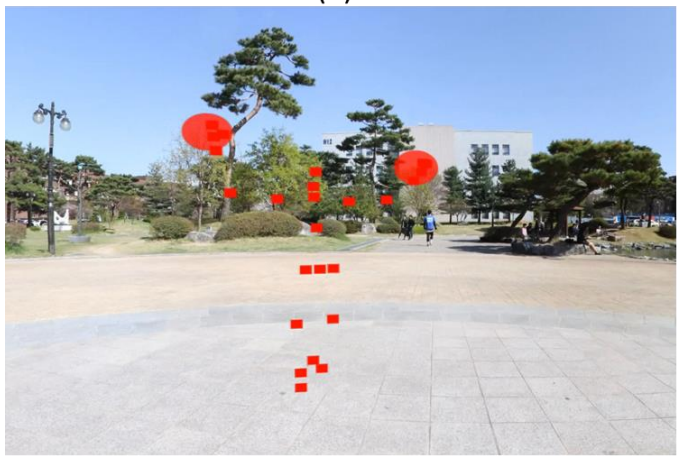
LAESkeleton representation



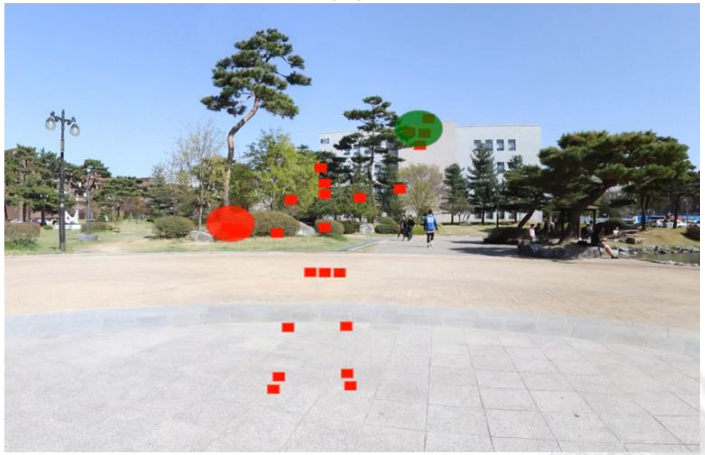
(a)



(b)



(c)



(d)

LAESkeleton representation and movement in 360^{o} virtual reality

MAR Scene and LAE-MAR – *HTML5 nodes for LAE representation*

```
<MARScene id= "mar1" description= "MAR for LAE representation" width= "1920"
height= "1080">
        <MARObject></MARObject>
        <LAE-MAR></LAE-MAR>
</MARScene>
```

```
<LAE-MAR id= "lae1" description= "LAE Representation" visible= "true">
        <LAE></LAE>
</ LAE-MAR >
```

LAE-MAR consists of LAEModel, LAECapturer, LAESensor, LAETracker, etc. which are the components of LAE representation system in MAR scene. This node provides the capabilities of accessing and controlling whole LAE representation system.

LAEModel – *HTML5 nodes for LAE representation*

LAEModel is an abstract node that provides all functionalities of the LAE representation.

Common properties:

id="lae1" description="LAE model" type="LAE2D" visible="true" showFPS="true",

According to the objective, LAEModel is represented as one among 2D, 3D, depth, and skeleton objects. So, this node provides LAE2D, LAE3D, LAEDepth, and LAESkeleton.

LAE2D – *HTML5 nodes for LAE representation*

LAE2D : LAEModel *{*

SFNode	[in,out]	metadata	null
SFString	[in,out]	description	""
SFString	[in,out]	id	""
SFInt	[out]	LAE_ID	0
SFString	[out]	type	LAE.type.LAE2D
SFImage	[out]	imageData	0 0 0
SFFloat	[out]	imageDataSize	0
SFFloat	[in,out]	laeX	0
SFFloat	[in,out]	laeY	0
SFFloat	[in,out]	laeWidth	1980
SFFloat	[in,out]	laeHeight	1080
SFBool	[in,out]	visible	true
MFString	[in,out]	behavior	null //Call event function

}

LAEDepth – *HTML5 nodes for LAE representation*

LAEDepth : LAEModel *{*

SFNode	[in,out]	metadata	null
SFString	[in,out]	description	""
SFString	[in,out]	id	""
SFInt	[out]	LAE_ID	0
MFString	[in,out]	type	LAE.type.LAEDepth
SFImage	[out]	depthData	0 0 0
SFFloat	[out]	depthDataSize	0
SFFloat	[in,out]	laeX	0
SFFloat	[in,out]	laeY	0
SFFloat	[in,out]	laeWidth	512
SFFloat	[in,out]	laeHeight	424
SFBool	[in,out]	visible	true
MFString	[in,out]	behavior	null //Call event function

}

LAE3D – *HTML5 nodes for LAE representation*

LAE3D : LAEModel *{*

SFNode	[in,out]	metadata	null
SFString	[in,out]	description	""
SFString	[in,out]	id	""
SFInt	[out]	LAE_ID	0
MFString	[in,out]	type	LAE.type.LAE3D
SFNode	[in,out]	color	X3DColorNode
SFNode	[in,out]	coord	X3DGeometryNode
SFNode	[in,out]	texture	X3DTextureNode
SFBool	[in,out]	visible	true
MFString	[in,out]	behavior	null //Call event function

}

LAESkeleton – *HTML5 nodes for LAE representation*

LAESkeleton : LAEModel *{*

SFNode	[in,out]	metadata	null
SFString	[in,out]	description	""
SFString	[in,out]	id	""
SFInt	[out]	LAE_ID	0
MFString	[in,out]	type	LAE.type.LAESkeleton
MFVec3f	[in,out]	boneCoord	[]//Human body
MFVec3f	[in,out]	skinCoord	[]
SFFloat	[in,out]	laeX	0
SFFloat	[in,out]	laeY	0
SFFloat	[in,out]	laeWidth	512
SFFloat	[in,out]	laeHeight	424
SFString	[in,out]	drawMode	[]//Skeleton or Skin
SFBool	[in,out]	visible	true
MFString	[in,out]	behavior	null//Call event function

}

LAECapturer – *HTML5 nodes for LAE representation*

```
LAECapturer : MARCapturer {
  SFNode      [in,out]  metadata     null
  SFString    [in,out]  id           ""
  SFString    [in,out]  description  ""
  SFNode      [in]      camera       LAECamera
}
```

LAECamera – *HTML5 nodes for LAE representation*

```
LAECamera : LAECapturer {
  SFNode      [in,out]   metadata     null
  SFString    [in,out]   id           ""
  SFString    [in,out]   description  ""
  MFString    [in]       cameraType   "general" //Depth, General, 3D, 360 Camera
  MFString    [out]      devicesID    []
  SFNode      [in]       camera       LAECamera
  MFNode      [in, out]  lae          LAE
  SFBool      [in,out]   enable       true
}
```

LAEGeneralCamera – *HTML5 nodes for LAE representation*

LAEGeneralCamera: LAECamera {

SFNode	[in,out]	metadata	null
SFString	[in,out]	Id	""
SFString	[in,out]	description	""
SFFloat	[in,out]	fov	45.0
SFInt	[in]	framerate	20
SFBool	[in,out]	audio	false
MFString	[out]	audio	[]
SFString	[in,out]	resolution	"fullHD"//Resolution QVGA,VGA,HD,Full HD
SFFloat	[in,out]	aspectRatio	1.5
SFString	[in,out]	filter	"none" //blur, Grayscale, Invert, Sepia
MFString	[in,out]	cameraMode	"user" //User, environment, left, right
MFString	[out]	cameraSource	[]
MFString	[out]	audioSource	[]
MFString	[out]	audioOutputSource	[]
SFBool	[in,out]	enable	true

}

LAEDepthCamera – *HTML5 nodes for LAE representation*

LAEDepthCamera: LAECamera {

SFNode	[in,out]	metadata	null
SFString	[in,out]	id	""
SFString	[in,out]	description	""
SFString	[in,out]	cameraSDK	"kinectv2" //Kinectv2, Creative, RealScene
SFBool	[in,out]	audio	false
MFString	[out]	audio	[]
SFFloat	[out]	frameRate	0
SFFloat	[out]	latency	0
MFString	[out]	resolution	[]
SFBool	[in,out]	enable	true

}

LAE360Camera – *HTML5 nodes for LAE representation*

LAE360Camera : LAECamera {

SFNode	[in,out]	metadata	null
SFString	[in,out]	id	""
SFString	[in,out]	description	""
SFBool	[in,out]	audio	false
SFFloat	[out]	frameRate	0
MFString	[out]	resolution	[]
SFString	[in,out]	generatedImage	"equirectangular" //Equiretangular, dual-fisheye
SFBool	[in,out]	enable	true

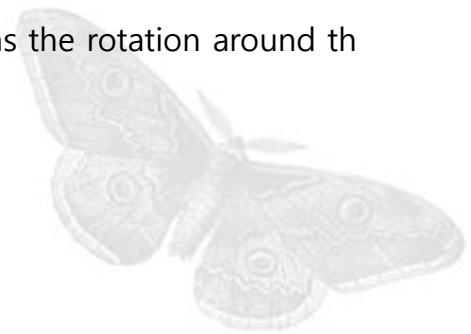
}

LAESensor – *HTML5 nodes for LAE representation*

```
LAESensor : MARSensor {
  SFNode      [in,out]  metadata      null
  SFString    [in,out]  id            ""
  SFString    [in,out]  description   ""
  SFString    [out]     sensorType    ""        //HMDsensor,
Joystick...
  MFString    [out]     sensorID      []
  SFBool      [out]     isConnected   false
  SFBool      [out]     hasPosition   false
  SFBool      [out]     hasOrientation false
  SFBool      [in,out]  enable        true
}
```

LAEControllerSensor – *HTML5 nodes for LAE representation*

```
LAEControllerSensor : LAESensor {
  SFNode        [in,out]  metadata          null
  SFString      [in,out]  id                ""
  SFString      [in,out]  description       ""
  SFString      [out]     sensorOutputType"" //Ratiometric
  SFVec3f       [out]     position          0 0 0 //Return the position along the [
x,y,z] axes.
  SFVec3f       [in]      rotation          0 0 0 //Return the rotation about the [
x,y,z] axes.
  SFVec3f       [out]     twist             0 0 0 //Returns the rotation around th
e Z axis.
  SFBool        [in,out]  vibration         false
}
```



LAETracker – *HTML5 nodes for LAE representation*

```
LAETracker : MARTracker {
  SFNode      [in,out]  metadata      null
  SFString    [in,out]  id            ""
  SFString    [in,out]  description   ""
  SFNode      [out]     LAECapturer   LAE.LAECapturer
  SFNode      [out]     LAESensor     LAE.LAESensor
  SFNode      [out]     LAE           null //LAE2D, LAEDepth, LAESkeleton
  MFString    [in,out]  trackingType  [] //face, body, environment tracking
}
```

LAEEnvironmentTracker – *HTML5 nodes for LAE representation*

```
LAEEnvironmentTracker : LAETracker {
  SFNode        [in,out]  metadata        null
  SFString      [in,out]  id              ""
  SFString      [in,out]  description     ""
  MFString      [in,out]  environmentType []  //Marker, multi-camera, natural
feature, incremental, outdoor tracking, SLAM
  MFString      [out]     environmentInfo []
  SFNode        [out]     LAE             null
  SFNode        [in,out]  LAECapturer     null
  SFNode        [in,out]  LAESensor       null
}
```

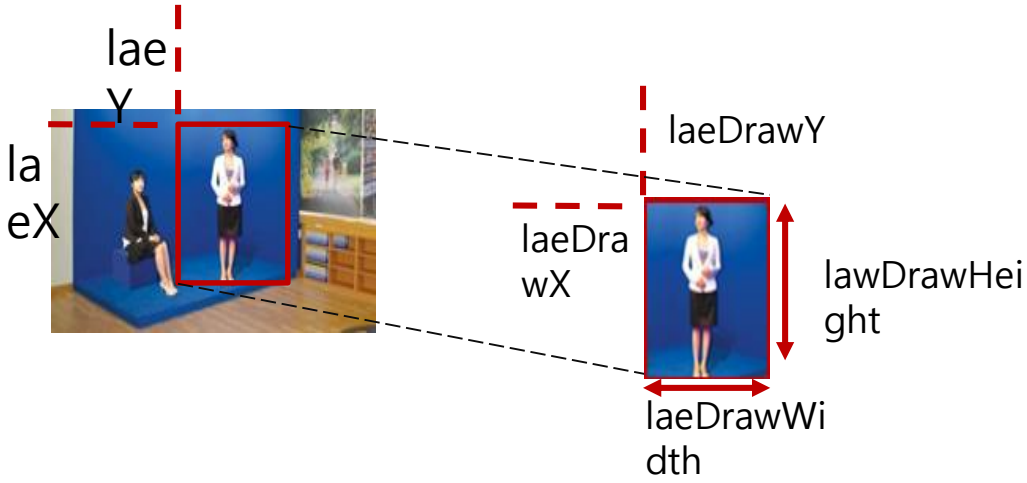
LAESpatialMapper – *HTML5 nodes for LAE representation*

LAESpatialMapper : MARSpatialMapper {

SFNode	[in,out]	metadata	null
SFString	[in,out]	id	""
SFString	[in,out]	description	""
MFVec3f	[in,out]	position	0 1 0
MFVec3f	[in,out]	direction	0 0 1
MFVec3f	[in,out]	scale	1 1 1
SFFloat	[in,out]	laeDrawX	0
SFFloat	[in,out]	laeDrawY	0
SFFloat	[in,out]	laeDrawWidth	0
SFFloat	[in,out]	laeDrawHeight	0
SFBool	[out]	isCollision	false
MFString	[in,out]	spatialInfo	[]
SFNode	[in,out]	LAETracker	null
SFNode	[in,out]	LAESceneRepresenation	null

}

LAESpatialMapper – *HTML5 nodes for LAE representation*



laeX	The x-coordinate, in pixels, of the upper-left corner of the ImageData object
laeY	The y-coordinate, in pixels, of the upper-left corner of the ImageData object
laeDrawX	Optional. The horizontal (x) value, in pixels, where to place the image on the canvas
laeDrawY	Optional. The vertical (y) value, in pixels, where to place the image on the canvas
laeDrawWidth	Optional. The width to use to draw the image on the canvas
laeDrawHeight	Optional. The height to use to draw the image on the canvas

LAERecognizer – *HTML5 nodes for LAE representation*

LAERecognizer : MARRecognizer {

SFNode	[in,out]	metadata	null
SFString	[in,out]	id	""
SFString	[in,out]	description	""
SFInt	[out]	eventID	0
MFString	[in,out]	eventInputType	[] //from joystick, hand gesture
MFString	[in,out]	eventType	[]
SFNode	[in,out]	LAECapturer	null
SFNode	[in,out]	LAESensor	null
SFBool	[in,out]	enable	true

}

LAEEventMapper – *HTML5 nodes for LAE representation*

LAEEventMapper : MAREventMapper {

SFNode	[in,out]	metadata	null
SFString	[in,out]	id	""
SFString	[in,out]	description	""
MFString	[out]	eventId	[]
MFString	[out]	eventDB	[]
MFString	[in,out]	eventHandling	[]
SFNode	[in,out]	LAERecognizer	null
SFNode	[in,out]	LAESceneRepresenation	null

}

LAESceneRepresentation – *HTML5 nodes for LAE representation*

```
LAESceneRepresentation : MARSceneRepresentation {
  SFNode      [in,out]  metadata     null
  SFString    [in,out]  id           ""
  SFString    [in,out]  description  ""
  MFString    [in,out]  sceneType    [] // 2D, 360VR, 3D scene
  SFBool      [in,out]  enable       true
}
```

Next Step

Submit it for CD

Thank you.