ARC Event Model

ISO JTC 1 SC 24 WG9

GERARD J. KIM
KOREA UNIVERSITY
Context: AR Events

- Initial approach (Extend X3D)
  - Extend X3DSensor Node for AR/MR events
  - Later: Make them usable in the Web/HTML too
  - Caution: Sensors mean “Events”

- Abstract X3DSensorNodes for AR: “X3DRealEnvSensor”
  - Marker
  - Image Patch
  - 3D Model
  - GeoLocation
  - Etc
AR/MR Content Model

- **Context**
  - Conditions for with augmentation to occur
  - “AR Events”
    - Marker recognition
    - Location recognition
    - ...

- **Augmentation**
  - 2D → HTML?
  - 3D → X3D?
  - Other: Haptic, Sound, ...

- **Context + Augmentation**
  - New constructs
  - X3D nodes
  - HTML elements?
What else?

- **Resources/Behavior (Augmentation)**
  - X3D Scene Elements (예: 3D objects)
  - HTML Elements (예: 2D objects)
- 기타
  - Device : Default 센서 이외의 사용자 정의 센서 및 속성
  - Background : Scene 이나 HTML Body/Canvas 의 Background 정의
AR Events and X3D/HTML Elements

- How to make the association?
  - New attributes
    - “update” attribute
      → Semantics: Every simulation round, update this node after processing the corresponding AR/MR event (and render)
    - “src” attribute
      → src : channel for accessing event data (e.g. transformation matrix)
  - Routes
  - Scripts / DOM (complicated behaviors only)
X3DRealEnvironmentalSensorNode : X3DSensorNode
{
SFVec3f [in,out] center 0 0 0 (-∞,∞)
SFBool [in,out] enabled TRUE
SFNode [in,out] metadata NULL [X3DMetadataObject]
SFVec3f [in,out] size 0 0 0 (-∞,∞)
SFTime [out] enterTime
SFTime [out] exitTime
SFBool [out] isActive
}

X3DEnvironmentalSensorNode is the base type for the real environmental sensor nodes
POI, 2D, 3D, ...
X3DPOISensorNode : X3DEnvironmentalSensorNode
{
    Type
    SFVec3f [in,out] location 0 0 0 (-∞,∞)
    SFVec3f [in,out] size 0 0 0 [0,∞)
    SFTime [out] enterTime
    SFTime [out] exitTime
    SFVec3f [out] centerOfRotation_changed
    SFBool [out] isActive
    SFRotation [out] orientation_changed
    SFVec3f [out] position_changed
}

Abstract representation for location or POI related events

Base type for GPS event, RFID event, etc.
GPSSensor : X3DPOISensorNode
{
SFVec3f [in,out] location 0 0 0 (-∞,∞)
SFVec3f [in,out] size 0 0 0 [0,∞)
SFTime [out] enterTime
SFTime [out] exitTime
SFBool [out] isActive
SFVec3f [out] position_changed
}

Synonymous/Interchangeable/mirrored with ARML style
<location type = GPS> construct
<RFIDSensor>

GPSSensor : X3DPOISensorNode
{
  X3DCoordinate Parent
  Integer TagInfo
  SFTime [out] enterTime
  SFTime [out] exitTime
  SFBool [out] isActive
}

Synonymous/Interchangeable/mirrored with ARML style
<location type = RFID ...> construct
X3D2DSensorNode : X3DEnvironmentalSensorNode
{
  image
  size
  center
  SFTime [out] enterTime
  SFTime [out] exitTime
  SFBool [out] isActive
  SFRotation [out] orientation_changed
  SFVec3f [out] position_changed
}

Abstract representation for 2D image recognition events
Base type for marker, image patch event, etc.
<MarkerSensor>

MarkerSensor : X3D2DSensorNode
{
  image
  size
  center
  SFTime [out] enterTime
  SFTime [out] exitTime
  SFBool [out] isActive
  SFRotation [out] orientation_changed
  SFVec3f [out] position_changed
}

Synonymous/Interchangeable/mirrored with ARML style

<marker ...> construct
**ImagePatchSensor**

```
ImagePatchSensor : X3D2DSensorNode
{
  image
  size
  center
  SFFloat [out] enterTime
  SFFloat [out] exitTime
  SFBool [out] isActive
  SFRotation [out] orientation_changed
  SFVec3f [out] position_changed
}

Synonymous/Interchangeable/mirrored with ARML style

<imagepatch ...> construct
```
<3DModelSensor>

3DModelSensor : X3DEnvironmentalSensorNode
{
  model
  size
  center
  SFTime [out] enterTime
  SFTime [out] exitTime
  SFBool [out] isActive
  SFRotation [out] orientation_changed
  SFVec3f [out] position_changed
}

Synonymous/Interchangeable/mirrored with ARML style

<3Dmodel ...> construct
CompositeSensor : X3DSensorNode
{
    condition expression
    values
}

Synonymous/Interchangeable/mirrored with ARML style
<composite_context ...> construct
<background>

- Defines what is drawn on the background of a given element such as canvas, scene, body, etc.

- `<background src="camo" type = video ... transparent="false">`  
  `</background>`

- Need to exactly define the semantics of “src” attribute or use additional attribute like type
  - Is it image? Is it video? Is it panorama
  - Or use Routes
There may be user-defined sensors with types and attributes not covered by the default model.

Devices are not used only for event detection or AR only.

```
<device id = "cam1" type = "camera" ... ></device>
```

Depending on the type, will have numerous optional parameters.

Could have sub nodes:
- Camera
- RFID
- GPS
- ...

Examples

<GPSSensor id = "context1" location = "38.234 45.567">
</GPSSensor>

<MarkerSensor id = "context2" image = "test/hiro.dat"
device = "cam1">
</MarkerSensor>

<3DModelSensor id = "context3" model = "test/model.3ds"
additional = "test/model.data"
device = "cam2"
...
</3DModelSensor>

<CompositeSensor context id = "context4"
condition = 'context1' OR 'context2'
value = 'context1.value + context2.value'
</CompositeSensor>
<!DOCTYPE HTML>
<html>
<head>
   <!-- some meta information -->
   <meta charset="UTF-8">

   <title>My first AR file format trial</title>

   <!-- use style sheet for x3dom (link is like include) -->
   <link rel="stylesheet" type="text/css" href="http://www.x3dom.org/download/x3dom.css">

   <!-- use x3dom script -->
   <!-- what does this script do? seems to be the whole library for x3dom functionality -->
   <script type="text/javascript" src="http://www.x3dom.org/download/x3dom.js">

   <!-- User defined sensor: a live camera -->
   <device id="cam0" type="WebCam" fov="50" framerate="30" />

   <!-- User defined sensor: a live camera -->
   <device id="cam1" type="WebCam" fov="50" framerate="30" />

   <!-- Location context -->
   <GPSSensor id = "context1" GPS_Location = "38.234 45.567"></GPSSensor>

   <!-- marker context -->
   <MarkerSensor id = "context2" 2Dimage = "test/hiro.dat"
               sensor = "cam1">
   
   </MarkerSensor>

</head>

</html>
<body>

<h1>AR/MR example 1</h1>

<!-- Play audio if context 1 is recognized by the browser-->
<audio update = "context1" controls="controls">
  <source src="test1.ogg" type="audio/ogg" />
  <source src="test1.mp3" type="audio/mpeg" />
  Your browser does not support the audio element.
</audio>

<!-- Play video if context 1 is recognized by the browser-->
<video width="320" height="240" controls="controls" update = "context1">
  <source src="giraff.ogg" type="video/ogg" />
  Your browser does not support the video tag.
</video>

<!-- Create and draw on this canvas if context 2 is recognized by the browser-->
<canvas id="myCanvas" width="200" height="100" update = context2>
  // context 2 에서 온 무언가가 canvas 내용에 영향을 미치게끔 함
  // This script simple draws a filled rectangle on the canvas if enabled by the context 2
  <script type="text/javascript">
    var c=document.getElementById("myCanvas");
    var ctx=c.getContext("2d");
    ctx.fillStyle="#FF0000";
    ctx.fillRect(0,0,150,75);
  </script>
</canvas>
</body>
Big Example

```xml
<canvas id="x3dcanvas">

<x3d xmlns="http://www.x3dom.org/x3dom" x="0px" y="0px" width="400px" height="400px">
// context 2에서 온 무언가가 scene 내용에 영향을 미치게 됨
<scene update="context2"> // first scene

<! background setting for this scene or canvas >
<background src="cam0" transparent="false"> </background>

// X3D scene graph specifics
<shape>
<appearance>
  <material diffuseColor='0.603 0.894 0.909'></material>
</appearance>
<box DEF='box'/> // this box is not augmented content, (just regular x3d element)
</shape>

<transform id="toViewPoint" translation="0 0 0", rotation="0 0 0"> <--- To --->
<Transform id="bySensorTracked" src="context2" device = cam1> <--- T1 --->
<Transform id="to_box" translation="0 0 0" rotation="0 0 0"> <--- T2 --->
<Shape>
  <Box size="4 4 4"/> // this box is augmented by context 2 with its context provided by context 2
</Shape>
</Transform>
</Transform>
</transform>

</scene>

Also use routes
</canvas>
```
// context 1 에서 온 무언가가 canvas 내용에 영향을 미치게끔 함
// Second scene in same canvas

<scene>

<background src="camo", transparent="false" ..... />

<Transform id="toViewPoint" translation="0 0 0", rotation="0 0 0" >

<Transform id="toText" translation="10 20 0" >

<Shape update = "context1"> // this example does not need src attribute

<Text data ="GPS 센서를 이용한 콘텐츠" size="4 4 4" /> // augmented text

</Shape>

</Transform>

</Transform>

</Scene>

</x3d>

</canvas>

</body>

</html>
Roadmap

• Work with Web3D AR Working Group
  ○ Finalize the information constructs
  ○ X3D extension proposal

• Others
  ○ AR Events
    ▫ Put forth as X3D independent constructs
    ▫ HTML 5 extension?
    ▫ ARML extension?
  ○ Event and Augmentation association
    ▫ Javascript
    ▫ Other attributes?