INTERNATIONAL STANDARD

ISO/IEC 14496-5

Second edition
2001-12-15

AMENDMENT 29
2011-11-01

Information technology — Coding of audio-visual objects —

Part 5: Reference software

AMENDMENT 29: Reference software for LASeR presentation and modification of structured information (PMSI) tools

Technologies de l'information — Codage des objets audiovisuels —

Partie 5: Logiciel de référence

AMENDEMENT 29: Logiciel de référence pour la présentation LASeR et la modification des outils d'information structurée (PMSI)

Reference number
ISO/IEC 14496-5:2001/Amd.29:2011(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 29 to ISO/IEC 14496-5:2001 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This Amendment introduces Reference Software for LASeR PMSI Tools. This Amendment deals with the reference software of ISO/IEC 14496-20:2008/Amd.3.

Information technology — Coding of audio-visual objects —

Part 5:
Reference software

AMENDMENT 29: Reference software for LASeR presentation and modification of structured information (PMSI) tools

At the end of Clause 5, add the following line:

Systems/LASeR_RS_v3/src	LASeR and SAF decoder for Presentation and Modification of Structured Information

At the end of B.3, add the following line:

Systems/LASeR_RS_v3/M3RP	LASeR and SAF renderer for Presentation and Modification of

Change Annex F to Annex G.

Change Annex G to Annex H.

Add the following new Annex F before Annex G:

Annex
(informative)

Guidance of reference software for LASeR presentation and modification of structured information tools

F.1 Introduction

This is the description of the reference software for ISO/IEC 14496-20:2009/Amd.3 (LASeR PMSI: Presentation and Modification of Structured Information). The LASeR AMD3 Reference software is attached to this Part of ISO/IEC 14496. The reference software consists of the following tools:

- LASeR decoder, which is to generate LASeR XML files from LASeR binary stream.
- LASeR file parser, which is to manage the scene tree and events from LASeR XML files.
- LASeR renderer, which is to render and play a LASeR scene.

The structure of LASeR PMSI reference software is as follows.

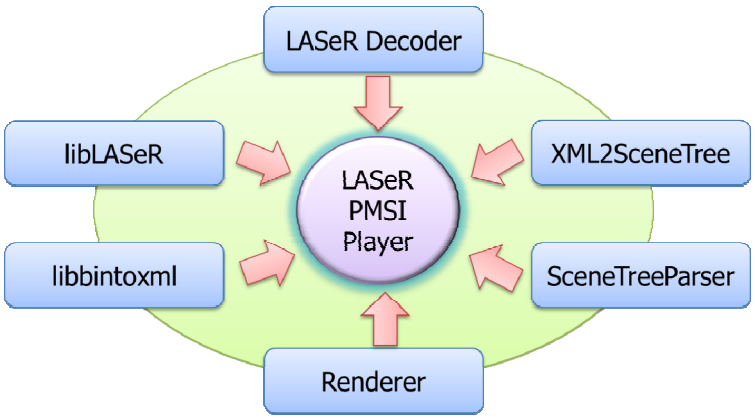


Figure F.1 — Structure of LASeR PMSI Reference Software

The LASeR player is linked with two static libraries such as libLASeR and libbintoxml, and the other dynamic ones such as LASeRDecoder, XML2SceneTree, SceneTreeParser and Render as shown in Figure F.1. By using these libraries, the LASeR player produces SceneTree from different file types such as xml, lsr files, and then, converts the SceneTree into a LASeR SceneTree. Finally, the player renders the scene according to the LASeR SceneTree.

F.2 Reference Software

The PMSI mechanism provides two distinctive functionalities for Structured Information (SI): presentation and modification. Therefore, the LASeR PMSI Reference Software can access SI from PI (Presentation Information) by two functionalities as follows:

- READ: Accessing SI from PI to reuse what is already defined in SI.
- WRITE: Updating SI from PI to reflect what happens in the scene.

Also, the elements provided by the LASeR PMSI Reference Software are as follows:

- **mpeg-pmsi() scheme**: Pointing scheme using XPath-based addressing.
- **SVG tref element:** Referencing the textual content
- **LASeR externalReference element:** Updating enclosed elements and attributes periodically.
- **LASeR externalUpdate element:** Modifying defined structured information.
- **LASeR xmlUpdate event**: Informing the update of structured information

F.3 Description of how to compile and run the LASeR PMSI Reference Software

F.3.1 How to build the code

This program has been developed using Microsoft's Visual studio 2008 under Windows XP or above OS. To build the program,

1) Unzip a LASeR_RS_v3.zip file.
2) Double-click on a M3R.sin file in the '\build \VC9' folder
3) Right button click on the LASeRPlayer in the solution explorer
4) Choose the 'Set as Start Up' menu
5) Click on the 'solution build' menu (F7)
6) Copy the files (libcairo-2.dll, libpng12-0.dll, zlib1.dll) in the '\dll' folder
7) Paste the files (libcairo-2.dll, libpng12-0.dll, zlib1.dll) in the '\bin \w32rel folder
8) Copy the files (libDeSAF.dll, libMP4.dll) in the '\dll \VC9' folder

Paste the files (libDeSAF.dll, libMP4.dll) in the '\bin \w32rel folder

F.3.2 How to run the code

To run the program,

1) Double-click on the LASeRPlayer.exe
2) Choose the 'File' menu
3) Click on the xml content (.xml) or the binary stream content (.lsr)

Figure F.2 shows the graphic user interface of LASeR PMSI Reference Software.

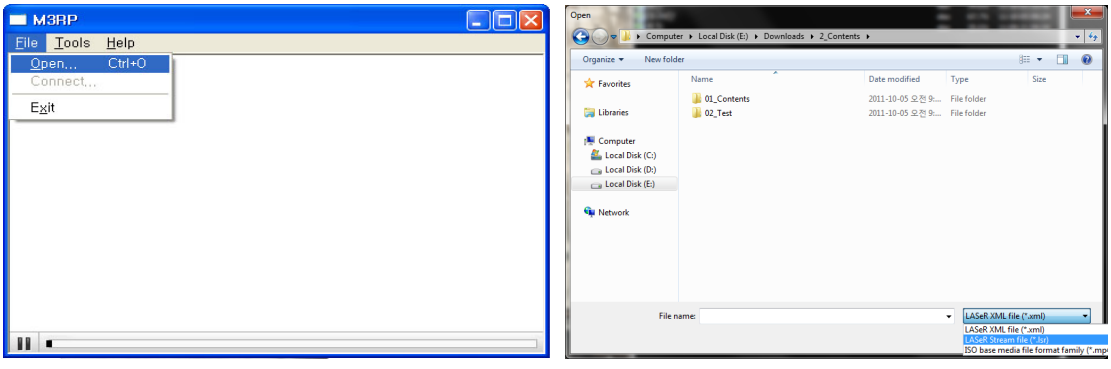


Figure F.2 — GUI of LASeR PMSI Reference Software

F.4 Demonstration

F.4.1 externalReference.lsr

Figure F.3 shows an example for referencing mechanism using LASeR externalReference element. It presents the local or remote metadata separately from the scene description in a scene.

LASeRML

```
<lsr:externalReference id="2" updateInterval="10">
  <text font-family="Tahoma" font-size="14" fill="rgb(0,0,0)" x="10" y="40">This is a PMSI 'externalReference' example.</text>
 - <text id="3" fill-opacity="1" font-family="Tahoma" font-size="20" fill="rgb(255,0,0)" x="10" y="80">
    <tref xlink:href="c:/Resource/test_DI.xml#pmsi(//mpeg7:FreeTextAnnotation/text())" />
  </text>
</lsr:externalReference>
```

Metadata – test_DI.xml

```
<Description xsi:type="CreationDescriptionType">
 - <CreationInformation>
   - <Creation>
      <Title type="main" xml:lang="ko">EBS</Title>
      <Title type="secondary" xml:lang="ko">1-subject</Title>
     - <TitleMedia>
       - <TitleImage>
          <MediaUri>c:/Resource/shrek.png</MediaUri>
        </TitleImage>
      </TitleMedia>
     - <Abstract>
        <FreeTextAnnotation xml:lang="en">This is the ISO/IEC 14496-20 AMD3 Ref. software</FreeTextAnnotation>
      </Abstract>
```

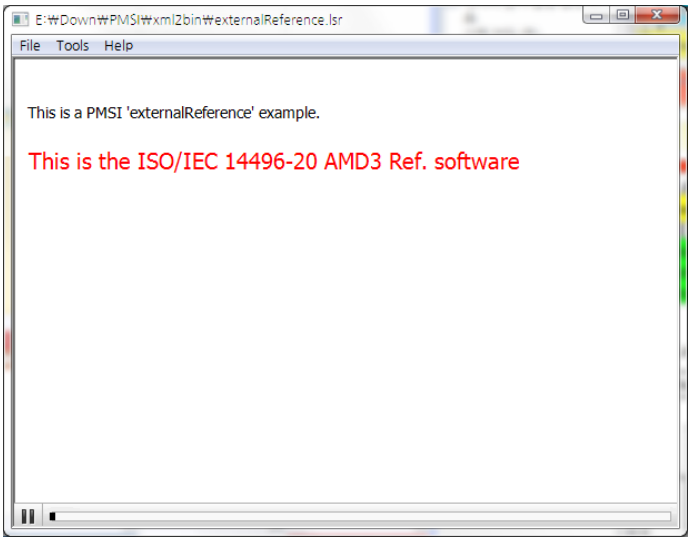


Figure F.3 — An Example of the LASeR externalReference element

F.4.2 externalUpdate.lsr

Figure F.4 shows the modification of the local or remote metadata using LASeR externalUpdate element.

- If the '1' or '2' keys are pressed, the referenced structured information will be changed to new one assigned to each key press action.
- If the '3' key is pressed, the referenced structured information will be removed.

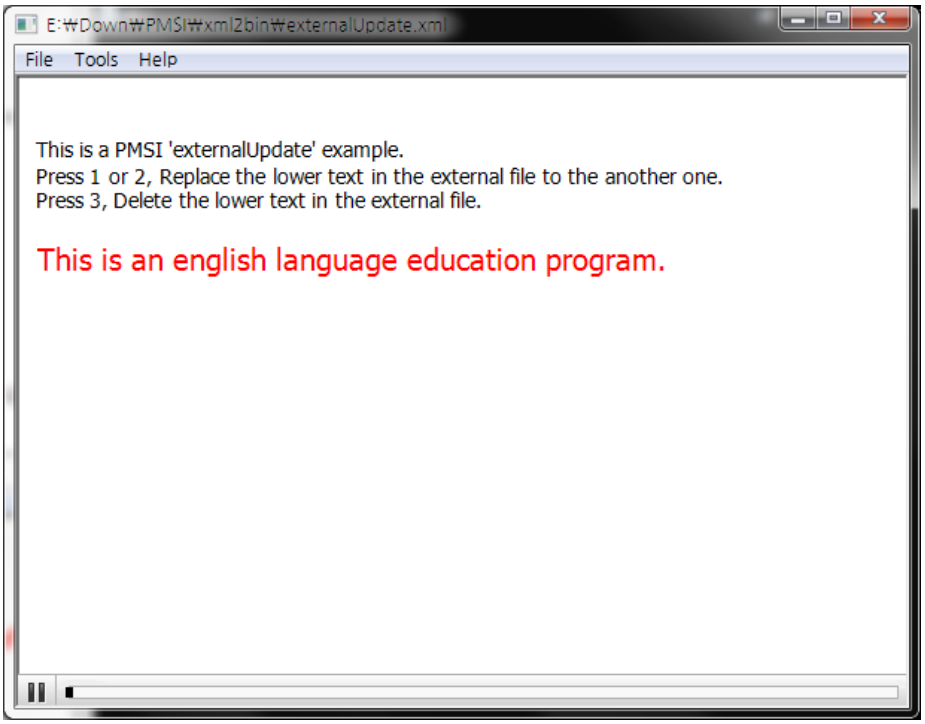
LASeR ML

```
<lsr:externalUpdate id="3" updateInterval="none" type="replace" attributeName="textContent" value="This is the ISO/IEC
    14496-20 AMD3 Ref. software" xlink:href="c:/Resource/test_DI.xml#pmsi(//mpeg7:FreeTextAnnotation/text())" />
<lsr:externalUpdate id="4" updateInterval="none" type="replace" attributeName="textContent" value="This is an english
    language education program." xlink:href="c:/Resource/test_DI.xml#pmsi(//mpeg7:FreeTextAnnotation/text())" />
<lsr:externalUpdate id="5" updateInterval="none" type="delete" attributeName="textContent"
    xlink:href="c:/Resource/test_DI.xml#pmsi(//mpeg7:FreeTextAnnotation/text())" />
</g>
<ev:listener event="accessKey(1)" handler="#3" enabled="true" />
<ev:listener event="accessKey(2)" handler="#4" enabled="true" />
<ev:listener event="accessKey(3)" handler="#5" enabled="true" />
```

(a) LASeRML

Metadata - The '1' key is pressed.

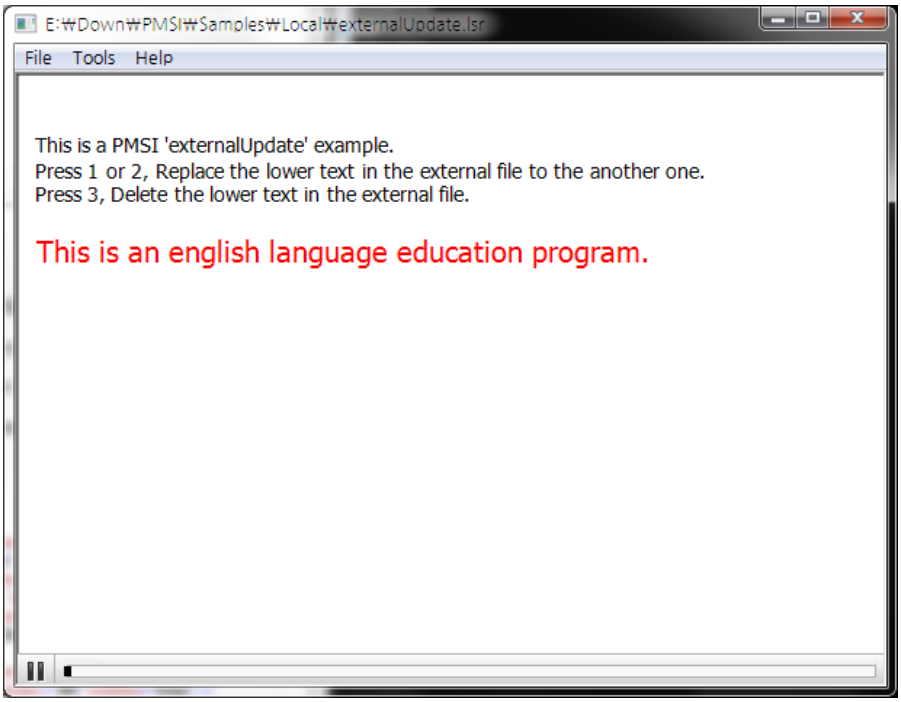
```
<Description xsi:type="CreationDescriptionType">
- <CreationInformation>
  - <Creation>
      <Title type="main" xml:lang="ko">EBS</Title>
      <Title type="secondary" xml:lang="ko">1-subject</Title>
    - <TitleMedia>
      - <TitleImage>
          <MediaUri>c:/Resource/shrek.png</MediaUri>
        </TitleImage>
      </TitleMedia>
    - <Abstract>
        <FreeTextAnnotation xml:lang="en">This is the ISO/IEC 14496-20 AMD3 Ref. software</FreeTextAnnotation>
      </Abstract>
```



(b) An example when the '1' key is pressed

Metadata - The '2' key is pressed.

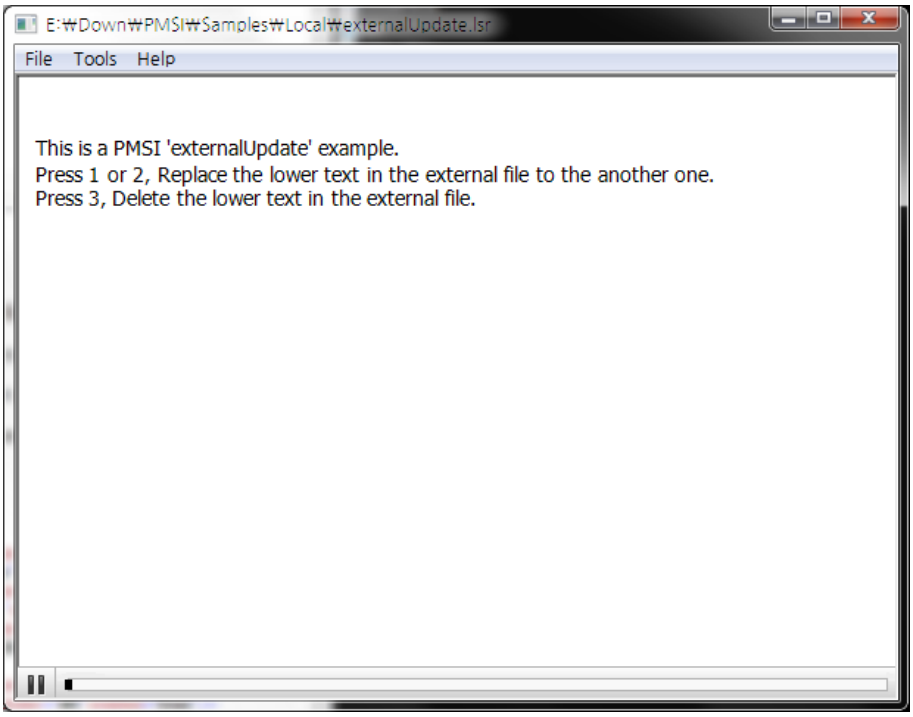
```
<Description xsi:type="CreationDescriptionType">
 - <CreationInformation>
   - <Creation>
      <Title type="main" xml:lang="ko">EBS</Title>
      <Title type="secondary" xml:lang="ko">1-subject</Title>
     - <TitleMedia>
       - <TitleImage>
          <MediaUri>c:/Resource/shrek.png</MediaUri>
        </TitleImage>
      </TitleMedia>
     - <Abstract>
        <FreeTextAnnotation xml:lang="en">This is an english language education program.</FreeTextAnnotation>
      </Abstract>
```



(c) An example when the '2' key is pressed

Metadata - The '3' key is pressed.

```
<Description xsi:type="CreationDescriptionType">
 - <CreationInformation>
   - <Creation>
      <Title type="main" xml:lang="ko">EBS</Title>
      <Title type="secondary" xml:lang="ko">1-subject</Title>
     - <TitleMedia>
       - <TitleImage>
          <MediaUri>c:/Resource/shrek.png</MediaUri>
        </TitleImage>
      </TitleMedia>
     - <Abstract>
        <FreeTextAnnotation xml:lang="en" />
      </Abstract>
```



(d) An example when the '3' key is pressed

Figure F.4 — An Example of the LASeR externalUpdate element

F.4.3 safRSH_image.lsr, safRSH_video.lsr

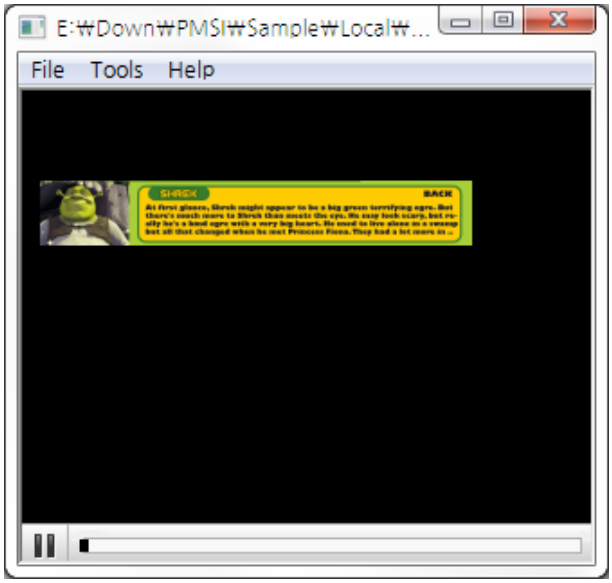
Figure F.5 shows the image or video rendering using the referenced metadata information.

LASeR ML

```
<saf:RemoteStreamHeader streamID="Stream1" objectTypeIndication="108" streamType="4" source="c:/Resource/test_DI.xml#pmsi
   (//mpeg7:TitleMedia/mpeg7:TitleImage/mpeg7:MediaUri/text())" />
<saf:sceneUnit>
 - <lsr:NewScene>
   - <svg id="1" height="240" viewBox="0 0 320 240" width="320">
     - <g>
         <image id="2" height="36" type="image/jpg" width="240" x="10" y="50" xlink:href="stream:Stream1" />
```

Metadata

```
<Description xsi:type="CreationDescriptionType">
 - <CreationInformation>
    - <Creation>
        <Title type="main" xml:lang="ko">EBS</Title>
        <Title type="secondary" xml:lang="ko">1-subject</Title>
       - <TitleMedia>
         - <TitleImage>
             <MediaUri>c:/Resource/shrek.png</MediaUri>
           </TitleImage>
        </TitleMedia>
```



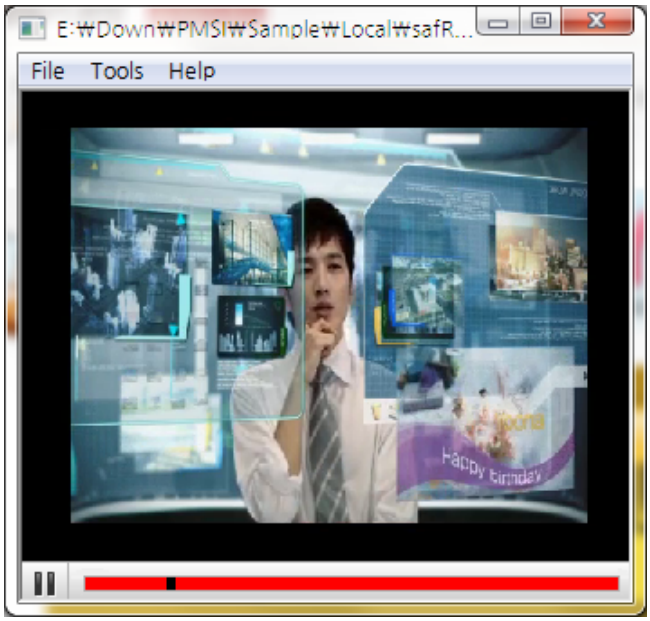
(a) Image rendering using PMSI

LASeR ML

```
<saf:RemoteStreamHeader streamID="Stream1" objectTypeIndication="33" streamType="4" source="c:/Resource/test_RV.xml#pmsi
   (//mpeg7:Target/mpeg7:Video/mpeg7:MediaLocator/mpeg7:MediaUri/text())" />
<saf:sceneUnit>
 - <lsr:NewScene>
    - <svg id="1" height="260" width="340">
       - <g>
           <video id="2" height="240" width="320" x="10" y="10" xlink:href="stream:Stream1" />
```

Metadata

```
<Description xsi:type="ViewDescriptionType">
- <View xsi:type="ResolutionViewType">
  - <Target>
    - <Video>
      - <MediaLocator>
          <MediaUri>c:/Resource/etri.mp4</MediaUri>
        </MediaLocator>
      </Video>
    </Target>
```



(b) Video rendering using PMSI

Figure F.5 — An Example of the pmsi scheme

F.4.4 xmlUpdate.lsr

Figure F.6 shows the demonstration for the LASeR xmlUpdate event. When the referenced metadata information is changed, the client updates the scene information.

LASeR ML

```
 - <text id="1" font-family="tahoma" font-size="20" fill="rgb(255,255,255)" x="10" y="80">
     <tref id="2" xlink:href="c:/Resource/test_DI.xml#pmsi(//mpeg7:FreeTextAnnotation/text())" />
   </text>
</g>
<ev:listener event="xmlUpdate" handler="#1" observer="2" enabled="true" />
```

(a) LASeRML

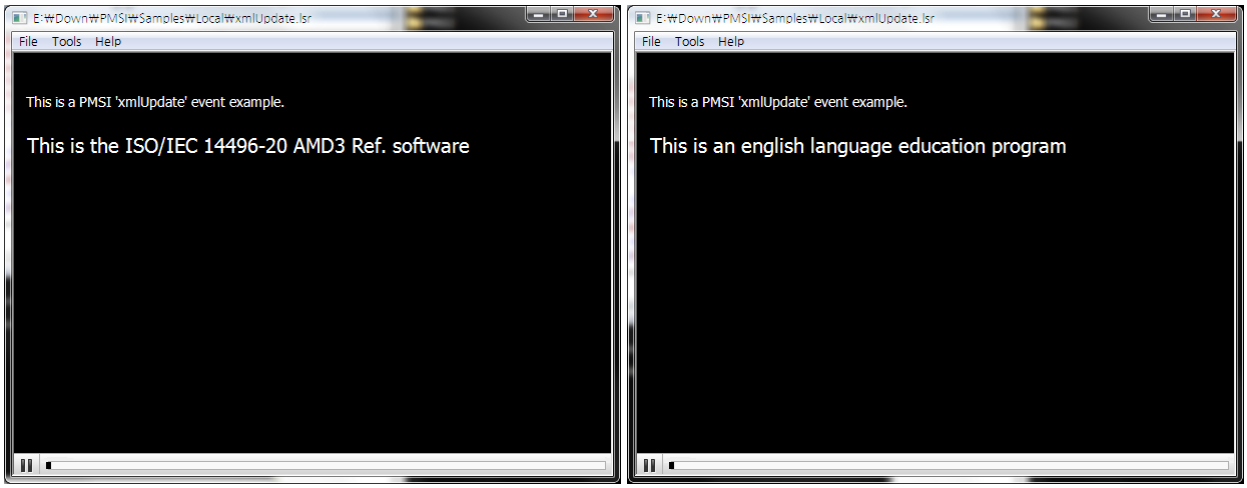
Metadata

```
<Description xsi:type="CreationDescriptionType">
 - <CreationInformation>
   - <Creation>
      <Title type="main" xml:lang="ko">EBS</Title>
      <Title type="secondary" xml:lang="ko">1-subject</Title>
     - <TitleMedia>
       - <TitleImage>
          <MediaUri>c:/Resource/shrek.png</MediaUri>
        </TitleImage>
      </TitleMedia>
     - <Abstract>
        <FreeTextAnnotation xml:lang="en">This is the ISO/IEC 14496-20 AMD3 Ref. software</FreeTextAnnotation>
      </Abstract>
```

(b) Orginal metadata

```
<Description xsi:type="CreationDescriptionType">
 - <CreationInformation>
   - <Creation>
      <Title type="main" xml:lang="ko">EBS</Title>
      <Title type="secondary" xml:lang="ko">1-subject</Title>
     - <TitleMedia>
       - <TitleImage>
          <MediaUri>c:/Resource/shrek.png</MediaUri>
        </TitleImage>
      </TitleMedia>
     - <Abstract>
        <FreeTextAnnotation xml:lang="en">This is an english language education program.</FreeTextAnnotation>
      </Abstract>
```

(c) Changed metadata



(d) Results when metadata is changed

Figure F.6 — An Example of the xmlUpdate event

F.4.5 epg.lsr

Figure F.7 shows an electronic program guide example using LASeR PMSI as a scene description and T-DMB metadata as structured information.

1) Click the 'myTV' button for EPG serivces as shown in Figure F.7(c).
2) Select the program that you want in the EPG menu. The detailed information will be displayed as shown in Figure F.7(d).
3) If you click the play button, the video will be started as shown in Figure F.7(e).

LASeR ML

```
<saf:RemoteStreamHeader streamID="Stream3" objectTypeIndication="108" streamType="4" source="c:/Resource/ETRI_PBS1.xml#pmsi
   (//dtva:ProgramInformation1/dtva:BasicDescription/dtva:MediaTitle/dmpeg7:TitleImage/dmpeg7:MediaUri/text())" />

<image id="9" fill-opacity="1" height="200" type="image/jpg" width="265" x="59" y="156" xlink:href="stream:Stream3">
  <set attributeName="visibility" begin="8.click" fill="freeze" to="visible" enabled="true" />
  <set attributeName="visibility" begin="10.click" fill="freeze" to="hidden" enabled="true" />
  <set attributeName="visibility" begin="11.click" fill="freeze" to="hidden" enabled="true" />
  <set attributeName="visibility" begin="12.click" fill="freeze" to="hidden" enabled="true" />
  <set attributeName="visibility" begin="13.click" fill="freeze" to="hidden" enabled="true" />
  <set attributeName="visibility" begin="14.click" fill="freeze" to="hidden" enabled="true" />
  <set attributeName="visibility" begin="15.click" fill="freeze" to="hidden" enabled="true" />
  <set attributeName="visibility" begin="16.click" fill="freeze" to="hidden" enabled="true" />
  <set attributeName="visibility" begin="17.click" fill="freeze" to="hidden" enabled="true" />
  <set attributeName="visibility" begin="18.click" fill="freeze" to="hidden" enabled="true" />
  <set attributeName="visibility" begin="5.click" fill="freeze" to="hidden" enabled="true" />
</image>
<textArea id="19" fill-opacity="1" font-family="Tahoma" font-size="18" font-weight="normal" fill="rgb(255,255,255)" height="47" width="171" x="596" y="187">
  <tref xlink:href="c:/Resource/ETRI_PBS1.xml#pmsi(//dtva:ProgramInformation1/dtva:BasicDescription/dtva:Title/text())" />
  <set attributeName="visibility" begin="3.click" fill="freeze" to="visible" enabled="true" />
  <set attributeName="visibility" begin="5.click" fill="freeze" to="hidden" enabled="true" />
  <set attributeName="visibility" begin="6.click" fill="freeze" to="visible" enabled="true" />
```

(a) LASeRML

Metadata

```
<dtva:ProgramInformation1 programId="01">
 - <dtva:BasicDescription>
    <dtva:Title xml:lang="en">My Pitiful Sister</dtva:Title>
   - <dtva:MediaTitle>
     - <dmpeg7:TitleImage>
        <dmpeg7:MediaUri>c:/Resource/PMSI1/ETRI_PMSI1_1.png</dmpeg7:MediaUri>
       </dmpeg7:TitleImage>
    </dtva:MediaTitle>
    <dtva:Synopsis xml:lang="en">A heartwarming family story unfolds, revolving around the eldest sister who makes sacrifices for
       her younger sisters. In-ok, the eldest of three sisters becomes the breadwinner of a poverty-stricken family at the age of
       thirteen when her parents passed away. Despite abject poverty, In-ok is always optimistic and confident, and lives her life to
       the fullest. Much to her great disappointment, however, she finds that her second youngest sister, In-su, has a baby out of
       wedlock. After giving birth to a baby girl, In-su leaves for Germany to pursue her dream of becoming a doctor, leaving the
       burden of raising her child to In-ok alone. This is a tribute to every woman in the world who has struggled to endure all
       hardships and taken good care of family by sacrificing her own life.</dtva:Synopsis>
   - <dtva:Genre href="urn:tva:metadata:cs:ProgramGenreCS:2006:3.1">
      <dtva:Name>Drama</dtva:Name>
    </dtva:Genre>
   - <dtva:ParentalGuidance>
     - <dmpeg7:ParentalRating href="urn:tva:metadata:cs:ParentalRatingCS:2006:3">
        <dmpeg7:Name>12세이상시청가</dmpeg7:Name>
       </dmpeg7:ParentalRating>
    </dtva:ParentalGuidance>
   - <dtva:RelatedMaterial>
     - <dtva:MediaLocator>
        <dmpeg7:MediaUri>c:/Resource/PMSI1/a2.mp4</dmpeg7:MediaUri>
       </dtva:MediaLocator>
    </dtva:RelatedMaterial>
    <dtva:ProductionLocation>kr</dtva:ProductionLocation>
  </dtva:BasicDescription>
</dtva:ProgramInformation1>
```

(b) Metadata

(c) Initial screen

(d) EPG using LASeR PMSI

(e) Video rendering

Figure F.7 — An Example of the electronic service guide using LASeR PMSI

F.4.6 ETRI_book.lsr

Figure F.8 shows an electronic book serivce example using LASeR PMSI. If you press the specific key, the related page can be offered as shown in Figure F.8 (c)-(e).

LASeR ML

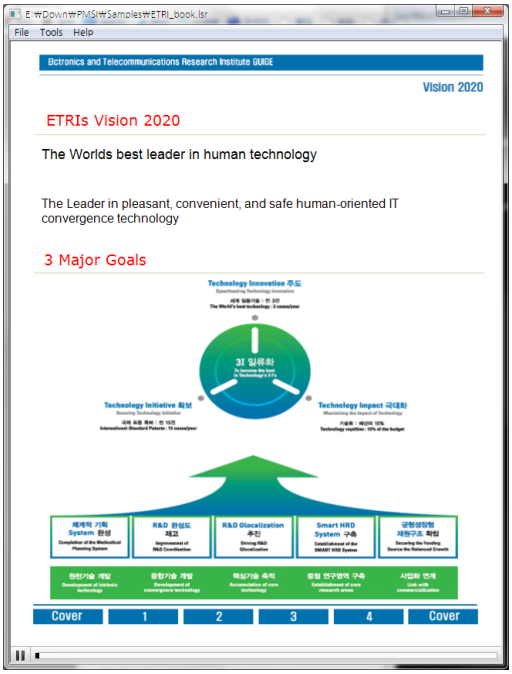
```
<lsr:externalUpdate id="9" updateInterval="none" type="replace" attributeName="textContent" value="ETRIs Vision 2020" xlink:href="c:/Resource/EbookMeta.xml#pmsi
    (//mpeg7:Title1/text())" />
<lsr:externalUpdate id="10" updateInterval="none" type="replace" attributeName="textContent" value="3 Major Goals" xlink:href="c:/Resource/EbookMeta.xml#pmsi
    (//mpeg7:Title2/text())" />
<lsr:externalUpdate id="11" updateInterval="none" type="replace" attributeName="textContent" value="The Worlds best leader in human technology"
    xlink:href="c:/Resource/EbookMeta.xml#pmsi(//mpeg7:FreeTextAnnotation1/text())" />
<lsr:externalUpdate id="12" updateInterval="none" type="replace" attributeName="textContent" value="The Leader in pleasant, convenient, and safe human-oriented IT
    convergence technology" xlink:href="c:/Resource/EbookMeta.xml#pmsi(//mpeg7:FreeTextAnnotation2/text())" />
<lsr:externalUpdate id="13" updateInterval="none" type="replace" attributeName="textContent" value="c:/Resource/Rbook/Rbg1.png"
    xlink:href="c:/Resource/EbookMeta.xml#pmsi(//mpeg7:TitleMedia/mpeg7:TitleImage/mpeg7:MediaUri/text())" />
<ev:listener event="accessKey(1)" handler="#9" enabled="true" />
<ev:listener event="accessKey(1)" handler="#10" enabled="true" />
<ev:listener event="accessKey(1)" handler="#11" enabled="true" />
<ev:listener event="accessKey(1)" handler="#12" enabled="true" />
<ev:listener event="accessKey(1)" handler="#13" enabled="true" />
```

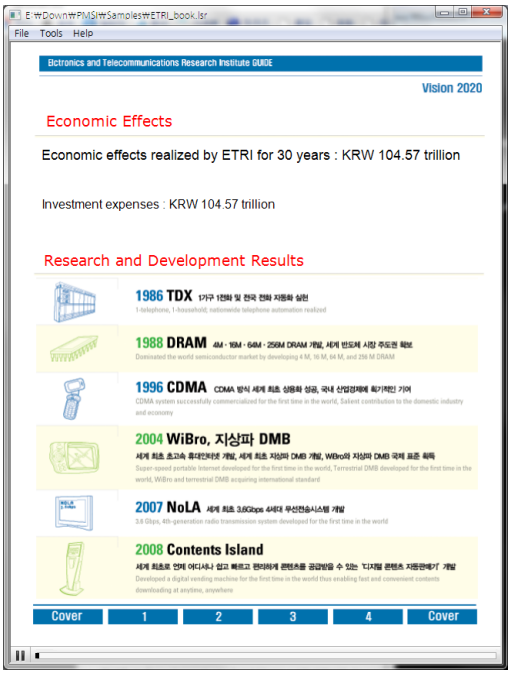
(a) LASeRML

Metadata

```
<Creation>
  <Title1 type="main1" xml:lang="en">ETRIs Vision 2020</Title1>
  <Title2 type="main2" xml:lang="en">3 Major Goals</Title2>
- <TitleMedia>
  - <TitleImage>
      <MediaUri>c:/Resource/Rbook/Rbg1.png</MediaUri>
      <MediaUri1>c:/Resource/Rbook/cover_end.png</MediaUri1>
    </TitleImage>
  </TitleMedia>
- <Abstract>
    <FreeTextAnnotation1 xml:lang="en">The Worlds best leader in human technology</FreeTextAnnotation1>
    <FreeTextAnnotation2 xml:lang="en">The Leader in pleasant, convenient, and safe human-oriented IT convergence technology</FreeTextAnnotation2>
  </Abstract>
```

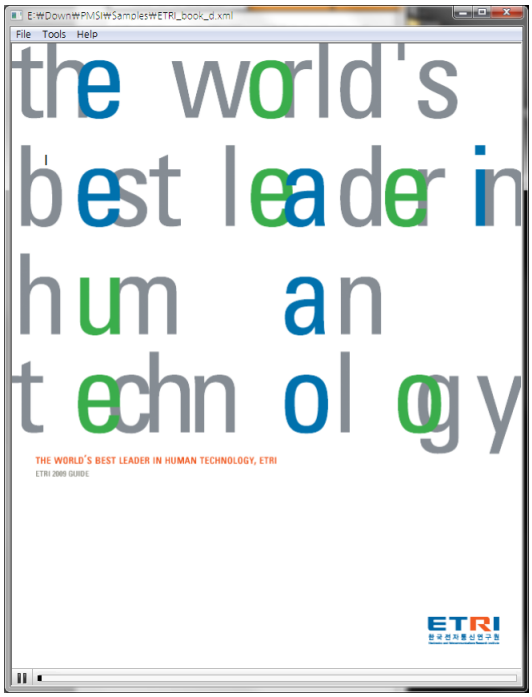
(b) Metadata





(c) '1' ~ '4' keys: page scroll

(d) '5' key: start rendering the video, '6' key: end rendering the video



(e) '0' key: return to cover page

Figure F.8 — An Example of the electronic book service using LASeR PMSI

F.5 The source code repository

Table F.1 — Directory description of the source code repository

Directory Name	Description
build\VC9	project (.vcproj) files that include the projects necessary for building the applications and solution file (.sln)
dll	necessary dll files
include	necessary header files
lib	necessary object file libraries
M3RP	code for the renderer
M3RP\LASeR	code for the renderer with LASeR elements
sample	informative sample conformance files
src\libbintoxml	code for the decoder
src\libLASeR	code for the parser
src\SceneTreeParser	code to parse a scene tree
src\XML2SceneTree	code to make a scene tree

F.5.1 libbintoxml

Table F.2 — Class description of libbintoxml

Class Name	Description
CA CAnimate …	class to store SVG and LASeR elements from the binary stream
CBIN2XMLManager	class to convert a binary stream file into a XML file
CBitReader	class to read a bit or byte value
CU CDNewscene …	class to store each command from the binary stream
CDecoder CPreDecoder	class to prepare each element for decoding
CIDManager CManager	class to manage the attribute for color, font, namespace, tag, and ID
CSAF_AU_Header CSAFHeader …	class to store a SAF header information from the binary stream
CScene	class to start decoding for scene
CSDLClasses	class of functions for a decoder
CURI	class to store the URI value information
CX CXpropElem …	class to store basic attributes from the binary stream

F.5.2 libLASeR

Table F.3 — Class description of libLASeR

Class Name	Description
CSVGObject CSVGProperties …	class to store SVG base properties in a scene tree
CActivate CAdd …	class to store LASeR commands in a scene tree
CSVGA CLSRConditional …	class to store SVG and LASeR elements in a scene tree
CLASeRHeader	class to store a LASeR Header
CESHeader CSAFUnit …	class to store SAF elements for processing SAF configurations
CLASeR	a base LASeR class for all derived classes
CSAF	a base SAF class for all derived classes

F.5.3 SceneTreeParser

Table F.4 — Class description of SceneTreeParser

Class Name	Description
CSceneTreeManager	class to manage a scene tree and provide convenient functions for processing LASeR commands

F.5.4 XML2SceneTree

Table F.5 — Class description of XML2SceneTree

Class Name	Description
CSceneCreator	class to create a scene tree using a DOM tree
CXMLFileManager	class to convert a XML file into a DOM tree using a XML parser

F.6 Used Open Source Library

- XMLite: simple XML parser (http://www.codeproject.com/KB/recipes/xmlite.aspx)
- Cairo: 2D graphical library (http://www.cairographics.org/)

ICS 35.040

Price based on 19 pages