

# MPEG-21 터미널

손유미<sup>1)</sup>, 박성준<sup>1)</sup>, Hendry<sup>1)</sup>, 정현<sup>1)</sup>, Qonita<sup>1)</sup>, 김종남<sup>2)</sup>, 박근수<sup>2)</sup>, 김문철<sup>1)</sup>

<sup>1)</sup> 한국정보통신대학교 멀티미디어 컴퓨팅, 통신 및 방송 연구실

<sup>2)</sup> KBS 기술연구소

e-mail : {yumi, mkim}@icu.ac.kr

## MPEG-21 Terminal

Yumi Sohn<sup>1)</sup>, Seongjoon Pak<sup>1)</sup>, Hendry<sup>1)</sup>, Hyun Jeong<sup>1)</sup>, Qonita<sup>1)</sup>, Jong-Nam Kim<sup>2)</sup>,  
Keunsoo Park<sup>2)</sup> and Munchurl Kim<sup>1)</sup>

<sup>1)</sup> Multimedia Computing, Communication and Broadcasting Laboratory,  
Information and Communications University (ICU)

<sup>2)</sup> Technical Research Institute, Korean Broadcasting System (KBS)

### 요 약

MPEG-21은 디지털 객체를 정의하고 이를 네트워크상에서 생성, 변형, 전달, 소비를 위한 통합적 멀티미디어 프레임워크에 대한 국제 표준이다. 디지털 아이템(DI: Digital Item)은 MPEG-21 프레임워크 내에서 표준화 된 식별체계, 서술 형식을 따르는 구조화된 디지털 객체이며, 유통, 처리의 최소 단위이다. 본 논문에서는 이러한 디지털 아이템을 최종 소비하는 MPEG-21 터미널 구조를 제안하고 이에 대한 설계 및 구현, 그리고 실험 결과를 제시한다.

## I. 서 론

우리는 다양한 종류의 멀티미디어 콘텐츠를 각기 다른 사용자 단말, 가변적 네트워크, 다양한 사용자 특성 등의 멀티미디어 환경에서 이용하고 있다. 이와 같이 각기 다른 사용자 터미널, 네트워크를 사용하는 다양한 멀티미디어 환경하에서 범용적 멀티미디어 접근을 위하여 전달과 소비를 위한 구성 요소들간의 상호 관련성을 기술하는 표준이 필요하게 되었다. 그런 표준이 MPEG-21이며 다양한 멀티미디어 콘텐츠를 다양한 환경에서 사용자가 상호 호환적으로 쉽고, 편리하게 생성, 배급, 소비할 수 있는 방법을 정의하고 구현 할 수 있도록 하는 '멀티미디어 프레임워크'를 구축하는 것을 목표로 한다.

MPEG-21 멀티미디어 프레임워크에서는 네트워크 상에서 디지털 데이터(콘텐츠)를 리소스라고 하며, 디지털 아이템(DI: Digital Item)은 멀티미디어 콘텐츠와 식별자, 그리고 서술자를 포함하는 유통, 처리의 최소 단위이다. DI는 단순한 단일 형태의 리소스 일 수도 있을 뿐만 아니라, 복잡한 형태의 복합적 멀티 리소스일 수도 있다. 그런 DI를 표준적으로 모델링 하기 위하여 디지털 아이템 정의 언어(DIDL: Digital Item Declaration Language)를 사용하여 정의하고, 이러한 DI가 사용자 단말에 전달되었을 때 규정된 절차대로 처리될 수 있도록 하기 위한 표준을 정하였는데, 이와 관련한 기술이 바로 디지털 아이템 처리(DIP: Digital Item Processing)이다.

본 논문에서는 이러한 디지털 아이템을 최종적으로 소비하는 MPEG-21 터미널 구조를 제안하고 이에 대한 설계 및 구현, 그리고 실험 결과를 제시한다. 이 모든 실험은 MPEG-21 멀티미디어 프레임워크 하에서, 디지털 아이템 처리를 고려한 테스트베드에서 이루어졌다.

## II. MPEG-21

### 1. Digital Item Declaration

MPEG-21 디지털 아이템 선언(DID: Digital Item Declaration)은 디지털 아이템의 구성요소 들을 통일되고 융통성 있는 개념의 상호 운용성 있는 구조로 기술하기 위한 규격을 정의하고 있다.

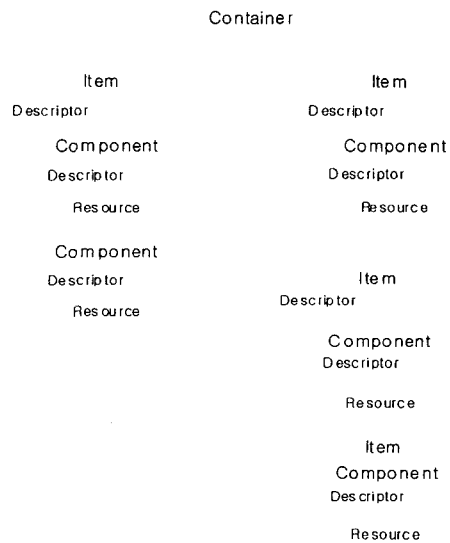


그림 1. DID 모델과 내부 구성 요소들간의 관계 [1]

DID의 목적은 디지털 아이템을 정의하는데 유용한 모델을 형성하기 위한 추상적인 개념 셋으로 기술하는 것이며 해당 아이템의 자원, 메타데이터, 상관 관계의 구체화를 포함하고 있다. 그림 1은 디지털 아이템 선언 모델의 계층구조와 모델 내부의 여러 주요 구성 요소들 간의 관계를 나타내고 있다. 여러 디지털 아이템을 복합적으로 포함하는 Container, MPEG-21 프레임워크의 거래, 전송, 이용될 수 있는 디지털 아이템의 기본 요소인 Item, 디지털 아이템의 부분 요소들을 표현하는 Component, 콘텐츠를 표현하는 Resource 등의 구성 요소를 사용하여 디지털 아이템을 선언할 수 있다.

## 2. Digital Item Adaptation

MPEG-21 디지털 아이템 적응(DIA: Digital Item Adaptation)은 디지털 아이템을 사용자 특성과 환경 정보, 네트워크나 터미널의 특성을 고려하여 다양한 멀티미디어 콘텐츠의 소비를 가능하게 하기 위한 범용적 멀티미디어 접근을 제공하기 위해 사용자 특성, 사용자 환경, 사용자 단말 특성, 네트워크 자원 특성 등에 대한 세부 정보를 체계적으로 기술하는 규격을 정의하고 있다.

그림 2는 MPEG-21 디지털 아이템 적응 변환의 개념도이다. 처리 과정은 기능적으로 크게 '리소스 적응 엔진'과 '서술자 적응 엔진'으로 분류 될 수 있는데, 사용자 특성과 사용 터미널 및 네트워크 환경 등의 사용(자) 환경 정보 등을 기술한 'DIA 서술자 틀'에 기반하여 입력된 디지털 아이템은 리소스 및 서술자 변환 과정을 거쳐 적응된(adapted) 디지털 아이템으로 출력된다. DIA 적응 틀은 디지털 아이템의 사용 환경을 서술하는 틀, 리소스 적응 틀, 디지털 아이템 적응 틀의 3개의 큰 틀로 구성된다.

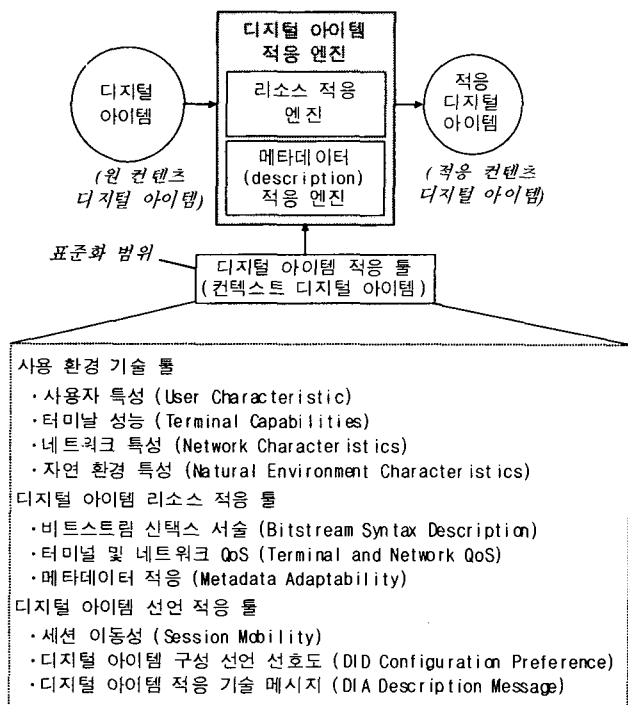


그림 2. 디지털 아이템 적응 변환의 개념 [1]

## 3. Digital Item Processing

디지털 아이템의 그 형태를 규정하는 표준은 이미 MPEG-21 DID에서 정의하고 있지만 이러한 DID는 콘텐츠의 내용 및 정보를 기술하는 정적인 형태이다. 따

라서 이러한 DID에 특정 동작을 처리할 수 있는 기능을 부여하고 그 기능들의 Processing 레벨에서의 상호 호환성을 규정하고자 하는 것이 DIP의 목적이다.

Processing 레벨에서의 상호 호환성을 정의하기 위한 DIP는 DID 문서를 받았을 때 DID의 요소들을 사용자에게 특정한 절차에 의해 표현하고 처리할 수 있는 방법에 대한 선택을 제공하는 Digital Item Method (DIM)과 DIM을 구성하는 Digital Item Base Operation (DIBO)의 기능과 역할을 정의하는 부분으로 이루어져 있다.

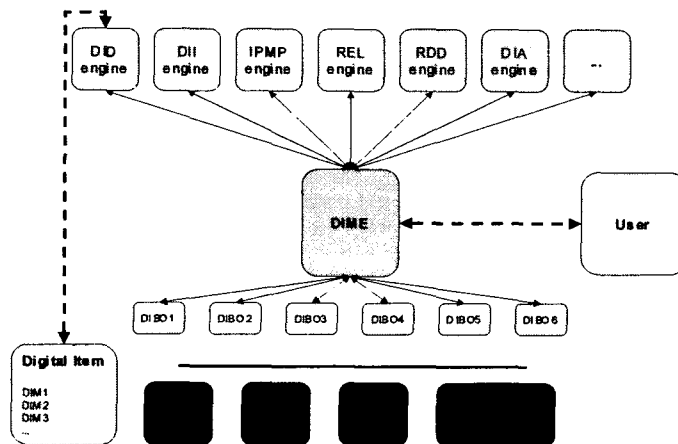


그림 3. MPEG-21 디지털 아이템 처리의 개관 [3]

그림 3은 MPEG-21 디지털 아이템 처리의 개괄을 표현하고 있다. DIM은 DID를 의도하는 형식이나 절차대로 처리하기 위하여 DIP Engine에 의해 수행되는 일련의 절차적인 동작들로 구성되어 DID 내에 포함되어 있는 DID 요소들을 처리, 식별하는 방법으로 조직화되어 있어야 하며, 발생하는 이벤트에 대해 처리할 수 있어야 한다. DI 문서가 터미널에 도착하면 먼저 DID Engine이 이를 처리하여 DIM을 인식하여 DIME에 보내고, DIME는 DIBO/DIM library에 접근하여 필요한 DIBO/DIM을 사용하거나 서버로부터 제공 받아 동작을 수행하게 된다.

## III. MPEG-21 Terminal Architecture

### 1. System Architecture

적용된 멀티미디어 콘텐츠 전달과 소비가 가능한 의미적 트랜스코딩을 위한 프레임워크를 만들기 위해서 우선 서버와 클라이언트로 구성된 시스템 구성도가 디자인 되어야 한다. 그림 4는 여러 종류의 클라이언트 터미널과 서버와의 관계를 나타낸 시스템 구성도이다.

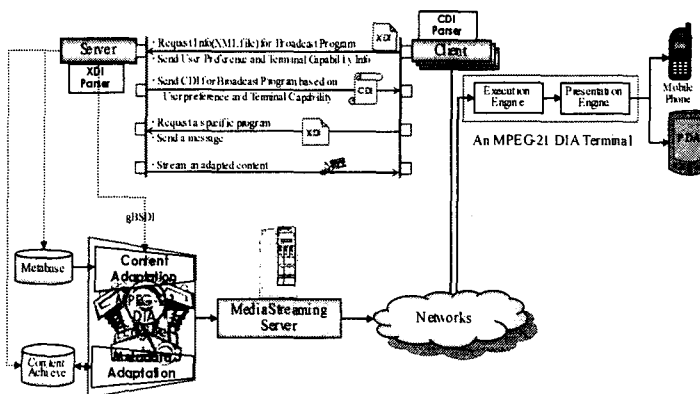


그림 4. 전체적인 시스템 구성도

클라이언트는 디바이스의 특성과 사용자 정보(환경 정보) 서술된 XDI (contexT Digital Item)를 서버 측에 보낸다. 서버는 XDI를 파싱한 후 클라이언트 환경에 맞는 멀티미디어 콘텐츠 정보가 기술된 CDI (Contents Digital Item)를 생성하여 클라이언트에 보낸다. 클라이언트에서 사용자가 특정 콘텐츠를 요청하면 이에 대한 정보 또한 XDI형식으로 서버에 전달된다. 이때 서버는 DIA 엔진을 통하여 리소스와 서술자가 적응변환된다. 연결된 리소스는 미디어 스트리밍 서버를 통해 적절한 형태로 사용자 단말에 전달되어 소비된다.

## 2. Terminal Architecture

그림 5는 DI를 소비하기 위한 MPEG-21 단말 구조를 나타내며 각 모듈에 대한 설명은 다음과 같다.

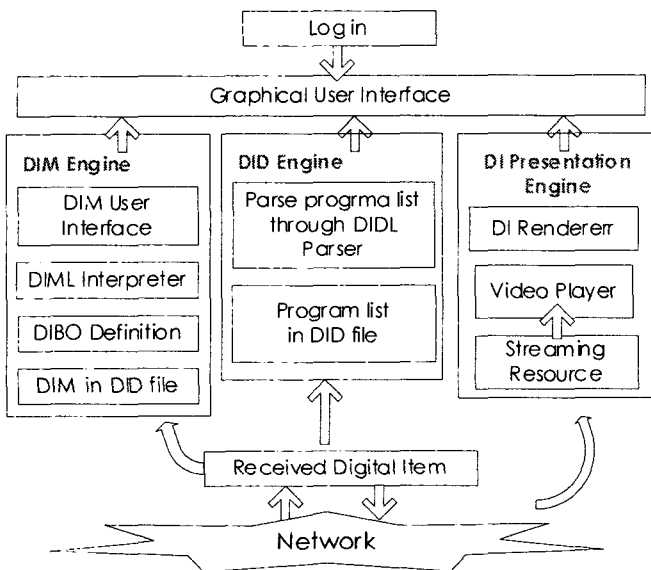


그림 5. MPEG-21 터미널 구성도

### A. DIM Engine

DIM은 네트워크를 통해 전달된 DID(CDI 포함)를 의도하는 형식 또는 절차대로 처리하기 위해 DIP Engine에 의해 수행되는 일련의 절차적인 동작들로 구성된다. DIM은 DID 내에 포함되어 DID 요소들을 처리, 식별하는 방법들로 조직화되어 있어야 하며, 발생하는 이벤트에 대해 처리할 수 있어야 한다. DIBO는 DI 내의 요소들의 display, 리소스와 어플리케이션 실행 등의 기능을 수행할 수 있는 기본 동작들을 의미한다.

### B. DID Engine

DID는 DIDL parser를 통해 파싱되고 DI에서 사용할 수 있는 DIM이 사용자 인터페이스에 나타난다. 사용자가 선택한 DIM은 DIML Interpreter에 의해 해석되고 DIM Engine의 DIBO와 연결되어 DIBO에 정의된 기능을 수행하게 된다.

### C. DI Presentation Engine

디지털 아이템을 표현하기 위하여 여러 응용 소프트웨어가 있을 수 있는데, 본 논문의 MPEG-21 터미널에서는 오디오 비주얼 콘텐츠를 재생하기 위하여 윈도우 미디어 플레이어를 기반으로 한 플레이어와 treeview와 slideshow를 디스플레이 하기 위한 패널을

이용하고 있다.

## IV. 실험

### 1. Digital Item

DIDL 파일에는 실제 디지털 아이템의 위치와 서술 등이 기술되어 있고 그 디지털 아이템들을 언제나 같은 형태로 처리되도록 하는 디지털 아이템 메소드를 기술한다. 아래 그림 6은 MPEG-21 터미널에서 쓰이는 DID 예제로 SOCCER란 Item에 대한 간단한 설명(description)과 사용하려는 리소스(참조 위치 포함)에 대하여 서술하고 있다. 하나의 Item 내부에 그 Item을 기술하기 위해 여러 세부 Item들(summary etc.)이 존재하고 그 Item들에 적용할 수 있는 2가지 메소드(Play, Treeview etc.)에 대하여 구체화하고 있다.

```
<Item id="DIM LIST">
  <Component id="PlayOriginalPC">
    <Descriptor>
      <Statement      mimeType="text/plain">
        OriginalVideo</Statement>
      </Descriptor>
      <Resource mimeType="text/diml"><![CDATA[
        function PlayOriginalPCwmv (item){
          var      Resource      =      GetDIDLNode
            ( "Descriptor[4]/Component[1]", item);
          Play (Resource)
        }]></Resource>
      </Component>
    <Component id="Treeview">
      <Descriptor>
        <Statement      mimeType="text/plain">
          OriginalVideo</Statement>
        </Descriptor>
        <Resource mimeType="text/diml"><![CDATA[
          function Treeview(itemList){
            var summaryList = didDocument.objectMap.
              getObjects(GetNodeID(itemList)+"SUMMARY_LIST");
            ShowTreeView(summaryList.length,
              GetDIDLNodeByID(GetNodeID(itemList) +
                "SUMMARY"));
          }]></Resource>
        </Component>
      </Item>
    <Item id="SOCCER">
      <Descriptor>
        <Component>
          <Resource mimeType="video/mpeg" ref=
            "http://210.107.133.71/MPEG21TB/worldcup/soccer
              .mpg" />
          </Component>
        </Descriptor>
        <Item id="SOCCER_SUMMARY">
          <Item id="SCENE1">
            <Descriptor><Statement      mimeType=
              "text/plain"> Kick      Off</Statement>
            </Descriptor>
            <Descriptor> <Component>
              <Resource mimeType="image/jpeg"
                ref="http://210.107.133.71/MPEG21TB/images/pkick
                  koff.jpg"/> </Component></Descriptor>
              <Component><Resource mimeType="video/wmv"
                ref="mms://210.107.133.71/MPEG21TB/worldcup/pkickoff.wmv"/></Compo
                nent>
            </Item>
            <Item id="SCENE2">
              <Descriptor><Statement mimeType=" text/
                plain">Shooting</Statement></Descriptor>
              <Descriptor><Component>
                <Resource mimeType="image/jpeg"
                  ref="http://210.107.133.71/MPEG21TB/images/psho
                    oting.jpg"/></Component></Descriptor>
                <Component><Resource mimeType="video/wmv"
                  ref="mms://210.107.133.71/MPEG21TB/worldcup/psh
                    ooting.wmv"/></Component>
              </Item>
            </Item>
          </Item>
        </Item>
      </Item>
    </Item>
  </Item>
</Item>
```

그림 6. DID 예제

DID는 DIDL parser를 통해 파싱되고 D에서 사용할 수 있는 DIM이 사용자 인터페이스에 나타난다. 사용자가 선택한 DIM은 DIML Interpreter에 의해 해석되고 DIM Engine의 DIBO와 연결되어 DIBO에 정의된 기능을 수행하게 된다.

### 2. DID 처리

MPEG-21 터미널 사용자의 요구에 따라 그에 맞는 DIDL(그림 6 참조)이 서버에서 터미널로 전달되면 파싱 과정을 거쳐야만 GUI상에 사용자가 가독할 수 있는 형태로 나타낼 수 있다. DID 엔진 모듈은 디지털 아이템과 디지털 아이템 메소드를 구분하고 DIDL parsing을 통하여 디지털 아이템을 디스플레이 하게 된다. 아래 그림 7은 DID 처리를 거친 PC와 PDA 터미널에서 콘텐츠 리스트를 보여주고 각 콘텐츠에 작용할 수 있는 디지털 아이템 메소드를 보여주고 있다. PC, PDA 2가지 디바이스 모두 Java를 사용하여 프로그래밍 하였다.

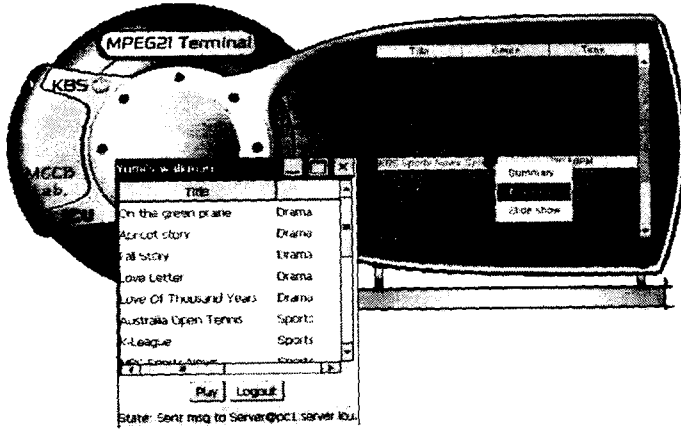


그림 7. DID 처리를 통해 표현된 PC & PDA GUI

### 3. DIM 처리

사용자가 DID처리를 거쳐 파싱된 디지털 아이템 메소드를 선택하게 되면 DIML(ECMA Script) 해석기를 이용하여 해당 DIM을 해석한 후 DIME 내부의 DIBO를 호출하여 해당 기능을 수행한다. 이를 통해 서로 다른 여러 종류의 디바이스에서 서비스 제공자가 정한 기능을 정확히 수행할 수 있다. 현재 테스트베드에서는 원형 콘텐츠와 비디오 요약 서비스를 제공하고 있는데, 그에 따른 DIM으로는 PlayOriginal, PlaySummary, Slideshow, Treeview 등이 있고 DIM 처리를 거친 예는 그림 8과 같다. Player는 Windows Media Player를 기본으로 하며 PC 버전에서는 SDK를 사용하였고, treeview와 slideshow는 Java로 프로그래밍 한 패널 위에 디스플레이 한다.

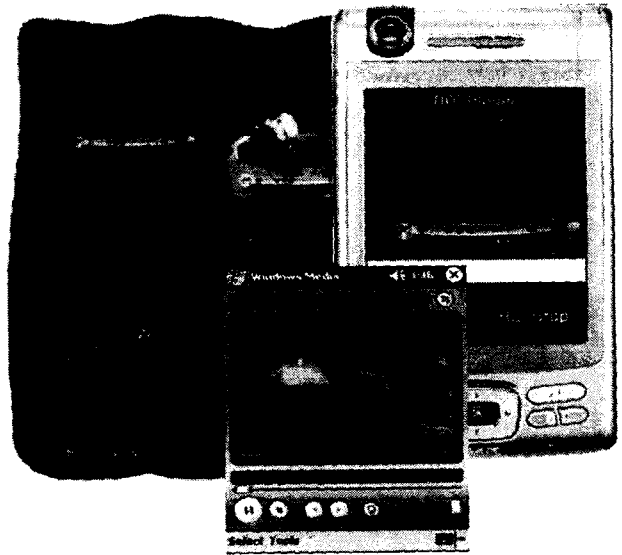


그림 8. DIM 처리를 거친 PC & PDA GUI

## V. 결론

본 논문에서는 MPEG-21 프레임워크에서 가장 기본적인 단위인 디지털 아이템이 실제로 각기 다른 사용자, 네트워크, 단말 등의 환경 하에서의 공통된 방법으로 사용될 수 있는 MPEG-21 통합 단말에 대한 연구와 구현 및 실험 결과를 제시하였다. 디지털 아이템의 전달 및 처리를 구체화 하기 위하여 네트워크 환경하에서 효과적인 콘텐츠의 소비를 가능하게 하는 비디오 요약 콘텐츠를 사용하였고 비디오 요약 서비스를 이용하여 DID, DIA와 DIP의 연동을 실제적 어플리케이션인 MPEG-21 터미널을 이용하여 실험하였다. 향후에는 MPEG-21 멀티미디어 프레임워크 내에서 DIP가 실질적인 역할을 수행하기 위해 MPEG-21의 다른 부분들 즉, RDD, REL, IPMP 등과 연동하여 수행할 수 있는 환경이 되어야 할 것이다.

### 참고문헌

- [1] ISO/IEC JTC1/SC29/WG11 N5231, " MPEG-21 Overview v.5" 62nd MPEG Shanghai, October 2002.
- [2] ISO/IEC 21000-7 Final Committee Draft, "MPEG-21 Digital Item Adaptation" ISO/IEC/JTC1/SC29/WG11 N5845, Trondheim, Norway, Jul. 2003.
- [3] ISO/IEC JTC1/SC29/WG11 N5855, MPEG-21 Digital Item Processing Working Draft (WD) v.2" Trondheim, Norway, Jul. 2003.