

웹 환경에서의 MPEG-21 테스트베드의 구현

MPEG-21 Testbed in WEB Environments

손정화, 손현식, 권혁민, 조영란, 김만배 · 강원대학교 컴퓨터정보통신공학과 영상통신연구실

Abstract

1990년대 후반부터 다양한 디지털 통신망을 이용하여 멀티미디어 콘텐츠 서비스가 가능하게 되었다. 하지만, 멀티미디어 콘텐츠의 전달 및 이용을 위한 기반 구조들의 독자적 발전 및 다양한 통합 관리 시스템 때문에 멀티미디어 콘텐츠 표현 방식의 호환성, 네트워크 방식과 단말기 등의 잠재적인 문제점이 발생한다. 이를 위해서 현재 존재하는 기술 및 기반 구조들 사이의 연동을 통한 큰 멀티미디어 프레임워크인 MPEG-21이 진행 중이다.

본 논문에서는 현재 표준화 작업이 진행 중인 MPEG-21을 기반으로 하는 웹(Web) 기반 테스트베드를 제안한다. 기본적으로 테스트베드는 서버(server), 클라이언트(client), DIA(Digital Item Adaptation)의 세 모듈로 구성된다. 서버의 역할은 멀티미디어 콘텐츠를 Digital Item(DI)의 형태로 생성하고, 클라이언트가 DI를 요구할 경우 DIA 모듈을 통해서 변환된 DI를 클라이언트에게 제공한다. DIA 모듈은 서버에서 동작되며 클라이언트로부터 요청된 DI를 분석하고 클라이언트로부터 전송된 환경 정보를 이용하여 클라이언트 환경에 적합하게 변환된(adapted) DI를 생성하는 것이 주 기능이다. 클라이언트는 서버에 저장되어 있는 DI를 선택하고

사용자 선호도(user preferences), 터미널 능력(terminal capabilities) 등의 필요한 정보를 서버로 전송한다. 테스트베드에서는 스포츠 경기의 동영상, 정지 영상, 경기 내용, 역사를 기록한 파일 등의 DI를 이용한다. 표현 언어는 XML이며, HTTP 기반의 웹 환경에서 구동되도록 설계된다.

1. 서론

1990년대 후반부터 다양한 디지털 통신망을 이용하여 멀티미디어 콘텐츠 서비스가 가능하게 되었다. 하지만, 멀티미디어 콘텐츠의 전달 및 이용을 위한 기반 구조들의 독자적 발전과 다양한 통합 관리 시스템으로 인해, 멀티미디어 콘텐츠 표현 방식의 호환성, 존재하는 네트워크 전달 방식, 단말 방식의 호환성 등의 잠재적인 문제점이 발생한다. 이것을 극복하기 위해서 현재 존재하는 기술 및 기반 구조들 사이의 연동을 통한 멀티미디어 프레임워크인 MPEG-21이 진행 중이다. MPEG-21은 표준화를 구체화하는 것부터 진행하여, 최종적으로 “다양한 네트워크 환경과 단말기에 있어서, 투명하고 통합적으로 멀티미디어 자원의 이용을 가능하게 하는 것”이다[1, 6].

MPEG-21은 Digital Item Declaration

(DID), Digital Item Identification and Description, Content Handling and Usage, Intellectual Property Management and Protection(IPMP), Terminals and Networks, Content Representation, Event Reporting 등의 7가지 요소로 구성되어 있다(1). 또한 표준화된 표현, 식별, 및 메타데이터(metadata)를 가지는 구조화된 객체를 디지털 아이템(Digital Item: DI)이라고 정의하였다(1, 2, 3).

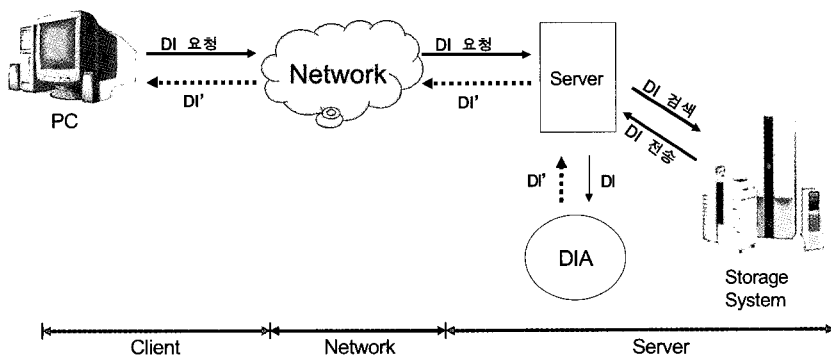
본 논문에서는 현재 표준화 작업이 진행 중인 MPEG-21을 응용하는 웹 기반 테스트베드를 제안한다. 테스트베드는 서버, 클라이언트, DIA(Digital Item Adaptation)의 세 모듈로 구성된다. 서버의 역할은 멀티미디어 콘텐츠를 디지털 아이템으로 생성하고, 클라이언트가 DI를 요구할 경우 DIA 모듈을 통해서 변환된 DI를 클라이언트에게 제공한다. DIA 모듈은 서버에서 동작되며 클라이언트로부터 요청된 DI를 분석하고, 필요하면 클라이언트로부터 전송된 환경 정보를 이용하여 적합하게 변환된 DI(adapted DI)를 생성한다. 클라이언트는 테스트베드에 저장되어 있는 DI를 선택하는 기능과 사용자 선호도(user preferences), 터미널 능력(terminal capabilities) 등의 필요한 정보를 서버로 요청한다. DI의 종류는 스포츠 경기의 동영상, 정지 영상, 경기 내용,

역사를 기록한 파일 등으로 구성된다. 또한 표현 언어는 XML이며, HTTP 기반의 웹 환경에서 구동되도록 설계된다.

본 논문의 구성은 다음과 같다. 2 절에서는 MPEG-21 테스트베드의 구조를 설명하고, 3 절에서는 서버와 클라이언트, DIA 모듈의 세부적인 구조와 기능을 서술한다. 4 절에서는 제안하는 테스트베드의 구조를 자세히 설명하며, 마지막으로 5 절에서는 본 논문의 요약 및 향후 과제에 대해서 언급한다.

II. MPEG-21 테스트베드 구조

본 논문에서 제안하는 MPEG-21 테스트베드의 구조는 [그림 1]과 같이 서버, 클라이언트, DIA 등의 세 모듈로 구성되어 있다. 서버는 DI의 생성, 수정, 삭제 등의 디지털 아이템 관리와 클라이언트로부터의 DI의 요청을 수신하며, DIA에 의해 적응된 DI를 다시 클라이언트에게 전송한다. 클라이언트는 웹 브라우저를 이용해서 서버에 있는 DI를 검색하고, 서버로 원하는 DI를 요청한다. 그리고 서버에서 보내온 DI를 웹 브라우저를 통해 소비한다. DIA는 서버에서 검색된 DI를 받아서 사용자 환경 정보(DIA Negotiation



(그림 1) MPEG-21 테스트베드 구성도

Message; DIANM)을 이용해 DI를 적응시키고, 이 적응된 DI를 다시 서버로 전송한다.

전체적인 흐름을 보면, 먼저 클라이언트는 디지털 아이템 브라우저를 통해서 서버에 있는 DI를 요청한다. 요청한 정보는 HTTP를 사용하는 네트워크를 통해 서버로 전송이 된다. 서버의 DI가 선택되어지면, 클라이언트는 DI를 소비할 때 서버의 DIA에서 사용되는 클라이언트의 환경정보를 정의한 DIANM를 서버로 전송한다. 서버는 네트워크로부터 수신된 DI 요청 정보를 얻는다. 서버는 서버가 관리하고 있는 DI에 대해서 요청한 정보를 일반 데이터베이스 관리 시스템을 사용하는 것이 아니라, XML 데이터를 파싱함으로써 검색한다. 데이터의 생성, 수정, 삭제 등의 관리 측면에서는 데이터베이스의 능력이 월등하지만, 테스트베드에서는 XML의 특징 중에 하나인 검색 기능을 이용한다[11]. XML에서 선언되거나 사용된 값들은 이를 파싱함으로써 실제적인 자료로써 이용되며, 검색 등과 같은 기능을 사용할 수 있게 된다.

서버는 클라이언트가 요청한 DI를 요청 정보와 함께 수신한 DIANM을 함께 DIA 모듈로 보내어 DI를 수정(적용)시킨다. 이렇게 수정(적용)된 DI

는 다시 네트워크를 통해서 클라이언트로 전송되고, 이를 수신한 클라이언트는 수정된 DI를 소비하게 된다.

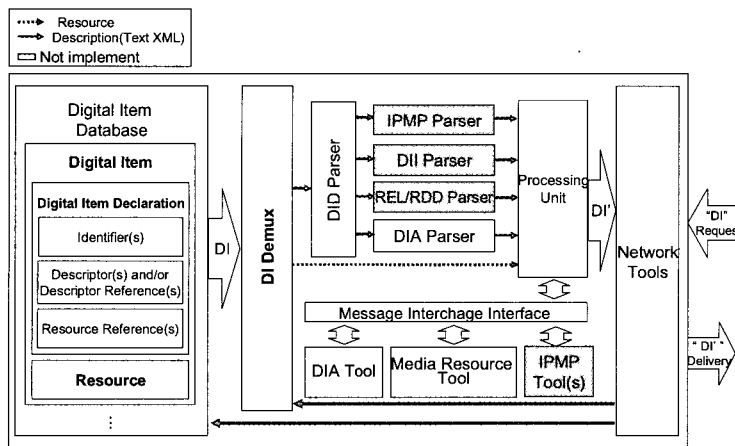
MPEG-21에서는 REL/RDD, IPMP 등과 같은 디지털 아이템의 지적권리에 대한 규약 등을 중요시 여기고 있지만[1], 테스트베드에서는 구현하지 않는다.

III. MPEG-21 테스트베드 모듈 구조

1. 서버

테스트베드의 서버는 [그림 2]처럼 *DI Demux, Parser, Message Interchange Interface, Tools, Processing Unit* 등의 크게 5가지 모듈로 구성된다.

DI Demux의 기능은 파서(parser)로 입력되는 DI 서술자(DI Description: DID)와 리소스(resource)로 분할하는 것이다. DID와 리소스가 이진(binary) 형태로 결합되어 있는 이진 DI인 경우에는 먼저 리소스와 DID 부분을 분리한 후 이진 DID를 텍스트 형태의 DID로 변환하여 DI



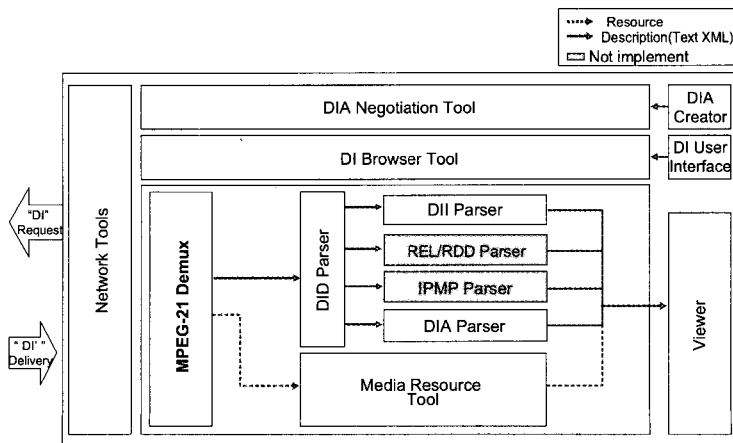
[그림 2] 서버의 구성도

를 분할한다. DID 파서는 DI로부터 분리된 DID를 각각의 파서가 파싱할 수 있도록 분할한다. 또한 DII 파서는 DID의 식별정보(Identifier Information Description)를 받아서 파싱한다. DIA 파서는 DID의 DIA 환경 정보와 클라이언트로부터 받은 DIA 협의 메시지를 파싱한다. 네트워크 파서(Network Parser)는 DID의 네트워크 식별 정보를 받아서 파싱한다. 메시지 교환 인터페이스(Message Interchange Interface; MII)는 툴들 간에 사용되는 정보의 공유 및 상호 운용을 가능하게 한다. 예를 들어, DIA 툴에서 리소스를 변경할 때 미디어 리소스 툴(Media Resource Tool)의 기능 중 디코딩(decoding)과 인코딩(encoding)을 이용하여 리소스를 변경한다. 처리 유닛(Processing Unit)으로부터 전달받은 서술자 정보와 리소스를 DIA 및 미디어 리소스 툴로 전달하고, 반대로 변경된 서술자 정보와 리소스를 처리 유닛으로 전달한다. DIA 툴은 클라이언트에서 보낸 DIA 협의 메시지(Negotiation Message)를 이용하여 요청한 DI를 수정한다. 미디어 리소스 툴은 DIA Tools에서 리소스를 변경하기 위해서 미디어 리소스들의 디코딩, 합성 및 렌더링을 수행한다. IPMP 툴은 IPMP 파서로부

터 받은 정보를 이용하여 IPMP의 역할을 수행한다. 네트워크 툴은 HTTP를 통하여 서버와 클라이언트의 상호 통신을 담당한다. 처리 유닛은 서버에서 파싱된 내용들과 DI Demux에서 출력된 리소스를 메시지 교환 인터페이스를 통해 DIA 툴과 상호 운용하여 클라이언트에 환경에 적용된 서술자와 리소스로 수정한 후 클라이언트에게 전송한다. 처리 유닛의 기능은 다음과 같다: (i) 파서에서 파싱된 서술자의 저장, (ii) DI Demux에서 출력된 리소스를 저장, (iii) 각 툴에게 메시지 교환 인터페이스를 통해 파싱된 서술자의 내용과 리소스를 전달, (iv) MII를 통해 얻은 툴에서 변경된 결과물들을 저장, (v) 클라이언트에게 전송한 수정된 DI를 생성, 및 (vi) 클라이언트에게 변형된 DI를 전송.

2. 클라이언트

클라이언트는 서버와 유사하게 DI Demux, Parser, Message Interchange Interface, Tool, Processing Unit 등으로 크게 5가지 부분으로 구성되어 있다. [그림 3]은 클라이언트의 블록도이다. 각 모듈들의 기능은 서버와 유사하다. Media



(그림 3) 클라이언트의 구성도

Resource Tool은 DIA Tools에서 리소스를 변경하기 위해서 멀티미디어 자원들을 처리한다. Network Tools은 테스트베드와 클라이언트간에 상호 통신을 담당한다. 즉, 클라이언트와 질의 화일, 응답 화일, DI 등을 송수신한다. DIA Negotiation Tool은 DIA Creator에서 클라이언트의 환경정보를 생성하면 이 정보를 XML 문서 형태로 변환한다. 이 정보를 네트워크 툴을 통해 서버로 전송을 되고, 서버는 이 정보를 이용해 DIA에서 DI를 수정한다. DI 브라우저 툴은 사용자가 테스트베드에 있는 DI를 선택할 수 있는 사용자 인터페이스를 제공한다. 일반적으로 인터넷의 경우 웹 브라우저의 기능을 하는 모듈이다.

3. DIA

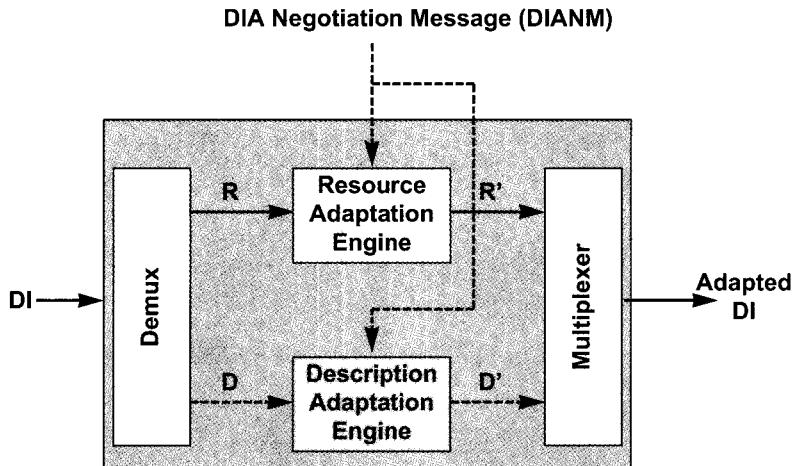
테스트베드의 DIA는 Demux, 리소스 적응 엔진, 서술자 적응 엔진, Multiplexer 등의 4 부분으로 구성되어 있다. (그림 4)는 DIA의 블록도이다. Demux는 서버와 동일하게 동작한다. 서술자 적응 엔진은 Demux에서 분리된 서술자와 클라이언트에서 보내온 DIANM을 이용하여 서술자

를 수정한다(10). 리소스 적응 엔진은 서술자 적응 엔진과 마찬가지로 Demux에서 분리된 리소스를 환경에 맞게 수정한다. 수정된 서술자와 리소스는 DI Multiplexer에 의해서 다시 DI로 구성된다. 테스트베드에서 DIA는 서버의 한 모듈로 수행된다. 클라이언트의 환경 정보는 사용자 선호도(user preference), 터미널 능력(terminal capability) 등을 고려한다(4,5,7).

IV. MPEG-21 테스트베드 구현

테스트베드를 구현하기 위해 먼저 사용 시나리오(Use Case Scenario)가 작성이 되어야 하고, 시나리오를 기반으로 콘텐츠와 메타데이터를 수집하여 디지털 아이템을 작성한다. 작성된 디지털 아이템은 XML과 마찬가지로 스타일시트에 의해서 웹 환경에서 사용자에게 제공된다. 그리고 시나리오에 의해 DIA 구현 기능도 같이 결정된다.

이 절에서는 위에서 언급한 테스트베드를 위한 시나리오, 테스트 디지털 아이템, DIA 구현, 그리



(그림 4) DIA의 구성도

고 웹 환경에서 사용자들에게 제공되는 사용자 인터페이스에 대해서 설명한다.

1. 사용 시나리오

본 논문에서 정한 사용자 시나리오는 다음과 같다. 스포츠를 좋아하는 사용자 A가 있다. A는 스포츠 중계를 보기 위해 사이트에 접속을 한다. 웹에 접속을 하였더니 4가지 스포츠를 주요 아이টে이션으로 하는 웹 페이지가 보인다. 아이টে이션은 축구, 농구, 배구, 하키 등 이다. A는 축구를 선택한다. 축구를 선택하였더니, 축구를 주제로 한 미디어들의 목록이 있다.

- 축구중계의 동영상과 메타데이터로 구성된 동영상 아이টে이션
- 축구중계의 음성만 저장한 오디오와 메타데이터를 가지고 있는 오디오 아이টে이션
- 축구경기의 장면과 메타데이터를 가지고 있는 영상 아이টে이션
- 축구경기의 역사를 저장하고 있는 문서에 메타데이터를 가지고 있는 문서 아이টে이션

사용자는 다음과 같은 과정을 이용하여 원하는 디지털 아이টে이션을 서비스 받을 수 있다.

- 동영상 View버튼을 클릭하니 화면에 MPEG-1 player와 table of contents (TOC)가 있다. A/V Player의 재생 버튼을 누르니, 서버에 있던 MPEG-1 동영상이 트랜스코딩(transcoding) 되어 서비스 된다. TOC에서 원하는 주요 장면, 즉 각 선수의 highlight를 선택하니, 동영상 화면이 원하는 장면으로 이동한다. Information을 클릭하면, 현재 경기에 대한 자세한 내용이 적혀 있는데, 자세한 내용의 목록은 방영시간, 선수명단, 감독, 심판 명단 등이다.

- Audio view버튼을 클릭하니, 서버에 있던 게임의 해설자가 설명했던 설명을 음성으로 저장한 오디오가 트랜스코딩되어 사용자 A에게 전달된다.

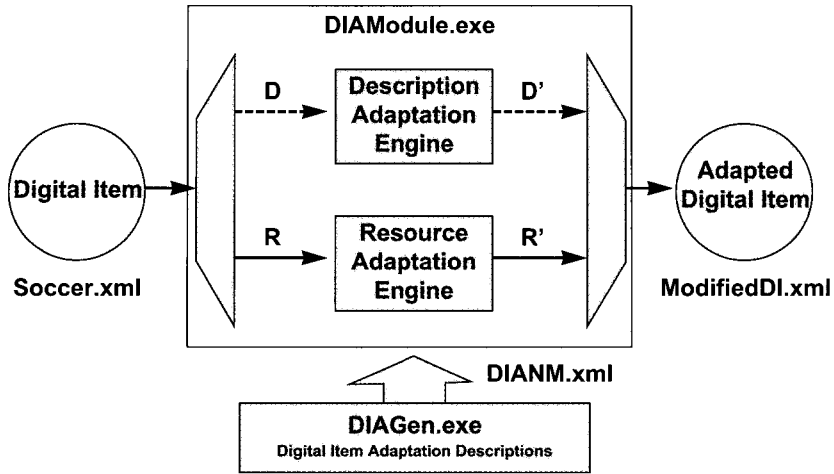
- Still Image view 버튼을 클릭하니, 서버에 있는 영상들의 목록이 보여진다. 첫 번째 목록의 View 버튼을 클릭하니, 해당하는 영상 및 영상 정보들이 서버로부터 트랜스코딩되어 서비스 된다.

- History View버튼을 클릭하니, 해당 종목의 역사와 유래, 경기 규칙을 기술한 문서가 사용자 A에게 서비스 된다.

2. 디지털 아이টে이션

테스트베드에서 사용되는 디지털 아이টে이션은 MPEG-21에서 규정한 DIDL에 의해서 정의된다[3]. 디지털 아이টে이션은 서술자와 컨테이너들에 의해서 묶여져 있다. 서술자는 이 디지털 아이টে이션이 어떠한 내용인지에 대해서 서술한다. 각각의 컨테이너는 아이디를 통해서 구분되어진다. 컨테이너는 서술자와 아이টে이션으로 묶여져 있다. 컨테이너 서술자는 컨테이너에 대한 설명을 기술한다. 아이টে이션은 공통적으로 서술자와 컴포넌트로 구성되어져 있다. 서술자는 아이টে이션이 가지고 있는 내용에 대해서 서술하고, 컴포넌트는 다시 서술자, 메타 서술자, 리소스로 구성된다. 서술자는 컴포넌트에 대한 설명을, 리소스는 디지털 아이টে이션의 자원에 대한 경로를, 그리고 메타 서술자는 리소스의 생성정보 등과 같은 메타데이터들을 서술한다. AV 아이টে이션은 비디오 세그먼트 데이터를 다시 하위의 컴포넌트를 이용해서 묶어서 정의하였다.

실제 적용한 디지털 아이টে이션은 크게 A/V 아이টে이션과 오디오 아이টে이션, 화상 아이টে이션, 히스토리 아이



(그림 5) 테스트베드에서의 DIA 실행모듈의 구성도

템으로 나누어진다. A/V 아이템은 MPEG-7에서 정의된 메타데이터(8, 9)와 MPEG-1, MPEG-2 동영상 리소스로 구성하였다. 오디오 아이템의 리소스는 wave, mp3 이고, 영상 아이템의 리소스는 jpeg, bmp 이다.

3. Digital Item Adaptation

DIA의 구현은 크게 DIA Negotiation Message (DIANM)를 생성하는 DIAGen 모듈과 디지털 아이템을 적응된 디지털 아이템(adapted DI)로 변환하여 주는 DIAModule 모듈로 구성된다 (그림 5)는 DIA 실행 모듈의 구성도이다. DIAGen은 웹 환경에서 클라이언트로부터 환경 정보를 입력받아 MPEG-21에서 정의하고 있는 DIA 표준에 맞는 DIANM.XML이라는 파일을 서버에 생성한다. DIAModule은 DIAGen 모듈에서 생성한 DIANM.XML 화일과 서버에 있는 디지털 아이템 화일을 이용하여 적응된 디지털 아이템인 ModifiedDI.xml 이라는 파일을 생성한 뒤 이를 사용자에게 보여준다.

DIA는 사용자 특성, 터미널 능력, 네트워크 특성, 자연 환경 특성 중 사용자 특성과 터미널 능력만을 고려하였다. 사용자 특성 측면에서 테스트베드는 처리도의 속성 중 leftSight와 rightSight 값을 이용해 사용자가 디지털 아이템을 시청 가능한지 여부를 정의하였다 (그림 6). LeftSight와 rightSight값이 모두 100이면, 현재의 사용자는 시청이 모두 가능한 상태이고, 모두 0이면 보는 것이 불가능한 상태이다. 만약 사용자가 시청가능한 상태가 아니라면, AV 디지털 아이템을 서비스할 때 Audio만을 서비스 한다.

터미널 능력 측면에서는 코덱의 decode 정의부분과 InputOutput의 display 정의를 이용하였다. (그림 7)은 터미널 능력을 표현하는 XML 파일을 보여준다. 우선 코덱은 decoding에서 해당 터미널에서 decode가 가능한 미디어의 포맷을 정의하였다. 만약 서버에 있는 디지털 아이템의 포맷이 여기에 정의되어진 포맷과 다를 경우 DIA에서는 파일포맷을 바꾸어서 사용자에서 서비스 한다. 그리고 InputOutput의 Display에서 현재 터미널의 출력이 가능한 크기와 칼라당 비트수를

```

<UserCharacteristics>
<!-- ##### -->
<!-- definition of Visable      -->
<!-- ##### -->
  <Visual leftSight="100" rightSight="100">
    </Visual>
  </Accessibility>
</UserCharacteristics>

```

(그림 6) 사용자 특성을 정의하는 XML

```

<TerminalCapabilities>
  <Codec>
    <Decoding>
      <SystemFormat>
        <MPEG2>TS</MPEG2>
      </SystemFormat>
      <ImageFormat>JPEG</ImageFormat>
      <AudioFormat>
        TerminalCapabilities/Codec/Decoding/ImageFormat
      </AudioFormat>
      <VideoFormat>MPEG2</VideoFormat>
    </Decoding>
  </Codec>
  <InputOutput>
    <Display colorCapable="true" backlightLuminance="1"
      refreshRate="90" bitsPerPixel="24">
      <Resolution horizontal="800" vertical="600"/>
      <SizeChar vertical="" horizontal="" />
    </Display>
  </InputOutput>
</TerminalCapabilities>

```

(그림 7) 터미널 능력을 정의하는 XML

MPEG-21 Testbed Broadcast System

Broadcast Main Item List			
Broadcast Item	Broadcast Date	Match	Note
Soccer	2002-10-06 16:00	Korea:Japan	<input type="button" value="Search"/>
Basketball	2002-10-05 10:00	Korea:China	<input type="button" value="Search"/>
Volleyball	2002-10-04 18:00	Korea:Japan	<input type="button" value="Search"/>
Hockey	2002-10-05 10:00	Korea:China	<input type="button" value="Search"/>

(그림 8) 주 아이템을 선택하는 화면 (컨텐츠 테마 선택)

MPEG-21 Digital Item of Soccer

Digital Items	
AV contents와 meta data로 구성된 Item	<input type="button" value="Search"/>
Audio contents와 meta data로 구성된 Item	<input type="button" value="Search"/>
Still Image contents와 meta data로 구성된 Item	<input type="button" value="Search"/>
Soccer's History contents와 meta data로 구성된 Item	<input type="button" value="Search"/>
Detail information of game	<input type="button" value="Search"/>

(그림 9) 스포츠 메인 화면 (SoccerMain.xml)

Soccer Game A/V (2002. 10. 27)

AV Viewer

DIA Part AV File's Detail Info

Creation Information

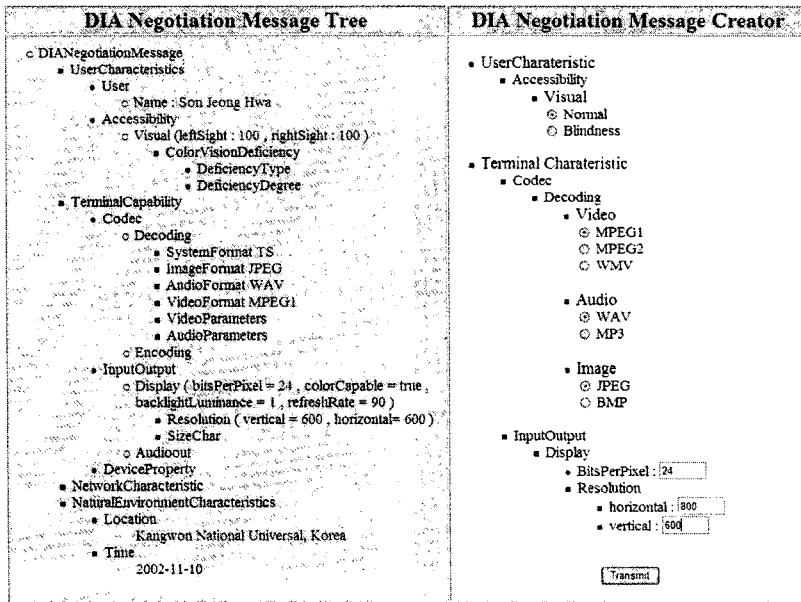
Content	Audiovisual
File Size	5,784,436 bytes
Format	MPEG1
Coding Format	MPEG-1 Video
Creator	Son Jeong Hwa
Date	2002.10.27 Mon.

Table of Contents

Heightlight	Frame
한국팀의 패널티킥	0
패널티킥을 위한 작전 지시	112
패널티킥 구간 표시	350
이관선수 모습	520
이관수 선수의 모습	600
연속되는 기회	700
종결	850

(그림 10) 동영상 디지털 아이템 서비스 화면

MPEG-21 DIA Negotiation Message



(그림 11) DIA Negotiation Message 전송화면

명시하여, 이미지를 서비스 받을 경우 정의된 사이즈보다 큰 이미지일 경우 사이즈가 조정되어서 사용자에게 서비스된다.

4. 구현

[그림 8]은 클라이언트가 서버에 접속하였을 때의 최초 서비스 되는 화면이다. 이 화면의 목적은 각각의 스포츠 아이템의 메인화면으로 이동하는 것이다. 여기에 적용된 stylesheet는 MainGUI.xml이다. 다음 단계로 진행하기 위해서는 각 줄의 View 버튼을 선택한다. [그림 9]는 각 스포츠의 세부 아이템 선택 화면으로, 세부 아이템은 동영상과 오디오, 정지영상, 역사를 기술한 문서 등으로 구성된다. 다른 단계로 진행하기 위해서는 서비스 받기 원하는 포맷을 선택하고 Search 버튼을 선택한다. 이 화면의 stylesheet는 SoccerMain.xml이다. 스포츠 종류와 서비스

받을 파일을 선택하면, 선택한 조건에 맞는 모든 디지털 아이템을 사용자에게 보여준다.

[그림 10]은 A/V 디지털 아이템을 서비스하는 화면이다. 화면은 크게 A/V Player와 Table of Content로 구성되어 있다. A/V Player의 재생버튼을 누르게 되면 서버로부터 컨텐츠가 서비스되며, Table of Contents의 Frame 번호를 클릭하게 되면, Highlight에서 설명하고 있는 장면으로 이동하여 재생한다.

최종 디지털 아이템을 선택하면, 서버에 있는 DIA에서 사용할 클라이언트의 환경정보를 전송하는 단계를 거친다. [그림 11]에서 이 단계는 크게 두개의 화면으로 사용자에게 서비스 된다. 좌측 화면은 이전에 서버가 사용한 DIA Negotiation Message (DIANM)를 의미 하며, 우측 화면은 실제 클라이언트가 수정하기를 원하는 내용을 선택하여 서버로 전송하는 기능을 한다. 클라이언트가 입력한 정보는 Transmit를 선

택함과 동시에 서버에 있는 DIA 모듈이 실행된다. DIA모듈은 DIAGen.exe와 DIAModule 프로그램을 통해 구현되었는데 이 프로그램을 실행하기 위해서 PHP 구문을 이용하였다. 이 구문에서 system은 유닉스의 'system' 명령어와 동일하게 동작되며, 인자로 사용하는 문자열의 내용을 서버에서 실행하는 기능을 한다. 두 실행 화일을 실행한 결과 생성되는 modifiedDI.xml은 마지막 echo 구문에 의해서 클라이언트에게 보여준다.

MPEG-21 기술들을 접할 수 있는 것을 증명하였다.

V. 결론 및 향후 연구

본 논문에서는 정보 교환을 위해서 일반화 되어 있는 인터넷을 위한 프로토콜인 HTTP 기반으로 서버-클라이언트 환경의 MPEG-21 테스트베드를 구현하였다. 테스트베드에서 서버는 디지털 아이템을 관리한다. 그리고 만약 클라이언트로부터 디지털 아이템에 대한 요청이 오면, 클라이언트로 디지털 아이템을 찾아 서비스한다. 또한 DIA를 이용하여 서버에 있는 디지털 아이템을 클라이언트가 수동적으로 서비스 받는 것이 아니라, 클라이언트는 자신의 환경 정보를 서버에 요청함으로써 서버로부터 디지털 아이템을 서비스 받을 때 클라이언트 환경에 적합하게 수정된 디지털 아이템을 받는다. 따라서 사용자는 디지털 아이템의 제약 없이 최적의 서비스를 얻을 수 있다. 그리고 테스트베드에서 사용된 디지털 아이템은 A/V, 오디오, 영상, 다양한 문서 등의 멀티미디어 데이터를 통합하는 수단으로 사용될 수 있었다.

테스트베드 구현에서 현재 표준화가 진행중인 MPEG-21의 요소 기술들이 웹 환경에서의 구현이 가능하고, 또한 구현을 통해서 일반 사용자가

저자 소개



손정화

2001년 2월 강원대학교 정보통신공학과 학사
2003년 2월 강원대학교 컴퓨터정보통신공학과 석사



조영란

2002년 2월 강원대학교 정보통신공학과 학사
2003년 강원대학교 컴퓨터정보통신공학과 석사과정 (현재)



손현식

2002년 2월 강원대학교 정보통신공학과 학사
2003년 강원대학교 컴퓨터정보통신공학과 석사과정 (현재)



김만배

1983년 한양대학교 전자공학과 학사
1986년 University of Washington 전기공학과 공학석사
1992년 University of Washington 전기공학과 공학박사



권혁민

2002년 2월 강원대학교 정보통신공학과 학사
2003년 강원대학교 컴퓨터정보통신공학과 석사과정 (현재)

1992년~1998년 삼성종합기술원 수석연구원
1993년 Georgetown University 의과대학 객원연구원
1996년 University of Rochester 전기공학과 객원연구원
1998년~현재 강원대학교 컴퓨터정보통신공학과 부교수
관심분야 : 비디오신호처리, 영상통신, 입체영상처리,

MPEG-21

■ 참고문헌

- [1] "MPEG-21 Overview V.4", ISO/IEC JTC1/SC29/WG11 N4801, May 2002
- [2] ISO/IEC Draft of Technical Reports 21000-1, "Part1: Vision, Technologies and Strategy", MPEG/N4333, July 2001.
- [3] "MPEG-21 Digital Item Declaration FDIS", MPEG/N4813, May 2002.
- [4] "MPEG-21 Digital Item Adaptation WD (v3.0)", MPEG/N5178, October 2002.
- [5] "MPEG-21 Digital Item Adaptation AM (v3.0)", MPEG/N5179, October 2002.
- [6] 김옥중, 홍진우, "MPEG-21 기술 개요", 제 4회 춘천 멀티미디어 합동 학술회의, 2002, 6월.
- [7] "MPEG-21 Requirements on Digital Item Adaptation", ISO/IEC JTC1/SC29/WG11 N4799, May 2002.
- [8] B. S. Manjunath, Philippe Salembier, Thomas Sikora, "Introduction to MPEG-7 Multimedia Content Description Interface", JOHN WILEY & SONS, LTD, 2002.
- [9] "MPEG-7 Part7: Multimedia Description Schemes", N4062, March 2001.
- [10] "MPEG-21 Digital Item Processing and DIA Negotiation Mechanism", ISO/IEC JTC1/SC29/WG11 M8516, July 2002.
- [11] W3C Recommendation, "Extensible Markup Language(XML) 1.0 (2nd Ed)," 6 October 2000.