

디지털 아이템 처리 표준화 동향

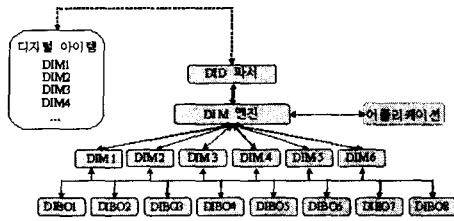
김문철 (한국정보통신대학교), 김종남, 박근수 (KBS기술연구소)

1. 서론

네트워크 환경 하에서 디지털 콘텐츠 사용자는 자신이 원하는 콘텐츠를 언제 어디서나 접근하여 자신이 원하는 형태로 소비가 가능하며, 콘텐츠 제공자는 디지털 콘텐츠의 생성, 관리, 전달 및 소비 단계에 이르는 모든 가치사슬(value chain)에서 제공 콘텐츠에 대해 언제 어디서나 식별 가능하고 추적 가능하며, 콘텐츠에 대한 권리 관리를 통한 안전한 콘텐츠의 유통을 가능하게 하기 위한 일환으로 추진 중인 MPEG-21은 디지털 객체(디지털 콘텐츠, 전자책 등의 디지털 데이터)를 리소스라 정의 하고 이러한 리소스에 대한 서술 데이터를 포함하여 디지털 아이템으로 정의하여 이를 네트워크 상에서 생성, 변형, 전달, 소비를 위한 통합적 멀티미디어 프레임워크 제공을 위한 국제 표준 기술이다.^[1]

디지털 아이템의 구성에 대한 선언 체계인 디지털 아이템 선언 (DID: Digital Item Declaration), 네트워크 상에서 디지털 아이템의 유일한 식별을 가능하게 하기 위한 식별체계를 제공하는 디지털 아이템 식별 (DII: Digital

Item Identification), 디지털 아이템에 대한 지적 자산 관리 및 보호 (IPMP: Intellectual Property Management and Protection), 디지털 아이템에 대한 권리를 명시하기 위한 언어인 권리 표현 언어 (REL: Rights Expression Language), 이와 관련한 권리 서술 사전 (RDD: Rights Description Dictionary), 네트워크 환경에서 범용적 멀티미디어 접근을 가능하게 하기 위한 네트워크 환경, 사용자 특성 및 사용자 터미널 특성에 대한 정보를 서술하기 위한 디지털 아이템 적응 (DIA: Digital Item Adaptation), 디지털 아이템이 소비될 때 특정 소비 형태를 정의하고 콘텐츠 제공자의 의도대로 제어하기 위한 디지털 아이템 처리 (DIP: Digital Item Processing), 다양한 터미널 및 가변적 네트워크 환경에서 리소스를 유연하게 표현하여 다목적용 단일 비디오의 압축/표현을 위한 스케일러블 비디오 부호화 (SVC: Scalable Video Coding), 리소스에 대한 식별자 정보 및 서술 정보를 리소스 내에 워터마크 형태나 또는 전송 포맷의 헤더 부분에 영속적으로 기록 해 두기 위한 기능을 정의하는 영속 관계 기술 (PAT: Persistent Association Technologies) 등에 대한 표준화 작



〈그림 1〉 Middleware의 구현 모듈

업을 진행하고 있다.

본 기고에서는 이러한 통합 멀티미디어 환경에서 디지털 아이템을 표준화된 형태로 상호호환성을 유지하기 위한 표준으로서 파트 10(ISO/IEC 21000-10)에 해당하는 디지털 아이템 처리에 대한 표준화 동향을 소개한다. 또한 본 표준 규격의 응용 사례에 대한 이해를 돕기 위해 본 논문의 저자들이 MPEG-21 표준화 활동을 통해 연구한 사례를 소개한다.

II. 디지털 아이템 처리의 개요

전술한 바와 같이 디지털 아이템 요소의 구성 및 관계를 서술하기 위해 MPEG-21 디지털 아이템 선언(DID)에서 정의하고 있으나, 이러한 형태로 서술된 디지털 아이템은 리소스의 내용 및 정보를 서술하는 정적인 형태에 지나지 않으며, 디지털 아이템이 사용자 터미널에 전달/전송 되었을 경우 이에 대한 처리 규정을 정의하지 않았다. 따라서 상호 호환적 형태로 디지털 아이템에 대한 처리 방식을 규정하고자 디지털 아이템 처리(Digital Item Processing)에 대한 표준화 작업을 시작 하였다.

디지털 아이템 처리 방식에 대해 상호 호환성을 정의하기 위해 디지털 아이템 처리는 디지털 아이템 기본 동작(DIBO: Digital Item Base

Operation) 인터페이스와 이러한 기본 동작 인터페이스를 이용하여 디지털 아이템의 처리 방식을 규정할 수 있는 디지털 아이템 메소드 언어(DIML: Digital Item Method Language), 디지털 아이템 메소드를 DID 문서 내에 바인딩 하기 위한 방법, 그리고 자바 언어를 이용하여 디지털 아이템 기본 동작 인터페이스를 바인딩 하기 위한 방법 및 디지털 아이템 메소드 내에서 디지털 아이템 확장 동작 인터페이스를 호출하는 방법에 대한 표준화 작업을 진행하고 있다²⁾.

디지털 아이템 메소드는 네트워크를 통해 전달된 DID 내에 포함되며, 리소스 제공자가 의도하는 형식 또는 절차에 따라 리소스를 처리하거나 DID 데이터를 조작하기 위해 일련의 절차적 동작들로 구성된 ECMA스크립트언어³⁾로 구성된 프로그램이다. 디지털 아이템 메소드는 DID 문서 내에 포함되어 DID 요소들을 처리, 식별하는 방법들로 조직화되어 있어야 하며, 발생하는 이벤트를 처리할 수 있어야 한다. 디지털 아이템 기본 동작 인터페이스는 디지털 아이템 내 요소들의 노드를 조작하거나, 리소스와 어플리케이션 실행 등의 기능을 수행할 수 있는 기본 동작들을 의미한다. 사용자가 선택한 DIM은 DIM 추출기에 의해 해당 DIM이 추출되어 디지털 아이템 메소드 엔진에 의해 실행된다. 이때 DIM엔진은 기본 동작 인터페이스를 호출하여 기능을 수행하게 된다.

그림 1은 사용자가 DID 파서를 거쳐 파싱된 특정 DID를 선택하게 되면 DIM 추출기를 이용하여 해당 디지털 아이템 메소드를 추출한 후, DIM 엔진을 통하여 미리 의도된 형태로 디지털 아이템이 처리된다. 이때 디지털 아이템 메소드가 특정 기본 동작 인터페이스를 포함하고 있는 경우, 디지털 아이템 메소드 엔진은 해

당 터미널 내부의 미들웨어로 구현된 기본 동작 인터페이스를 호출하여 기능을 수행한다. 이를 통해 서로 다른 여러 종류의 터미널에서도 디지털 아이템 제공자가 의도한 형태로 어떤 터미널에서도 표준화된 형태로 아이템이 처리되고 조작된다.

III. 디지털 아이템 처리 표준화

1. 디지털 아이템 메소드

DIP 기본 개념은 DID를 받을 때 디지털 아이템을 다양하게 표현하는 DIM 정보를 포함해서 받는 것이다. 사용자는 DIP 엔진에서 원하는 DIM을 선택함으로써, 원하는 형태로 디지털 아이템을 소비할 수 있다. DIM은 DID를 의도하는 형식 또는 절차대로 처리하기 위해 DIP Engine에 의해 수행되는 일련의 절차적인 동작들로 구성되며, DID 내에 포함되어 DID 요소들을 처리, 식별하는 방법들로 조직화되어 있어야 하며, 발생하는 이벤트에 대해 처리할 수 있어야 한다.

저자, 출판 제작자, 배급업자들이 디지털 아이템을 사용할 때 필요한 여러 가지 디지털 아이템의 처리 방법을 DIM에 기술함으로써, 사용자는 DIM을 이용해서 보다 다양하고 유연한 방법으로 콘텐츠를 표현할 수 있다. 예를 들면, 어떤 음악 앨범이 디지털 아이템으로 선언되었을 때, "AddTrack" 이라고 정의된 DIM은 사용자가 디지털 아이템에 새로운 음악을 더하는 명령을 수행할 수 있다. 그러나, DIM은 미디어 데이터 자체에 대한 처리를 수행하지 않는다. 예를 들면, DIA의 영역인 멀티미디어 데이터의 트랜스코딩은 DIM의 범주에 속하지 않는다.

DIM은 사용자 사용관점에서 정의되며, 주로 사용자와 상호 작용하는 기능에 관련되어 있다. DIM을 사용하기 위해서는 DID에서 DIM을 사용하기 위한 초기화 과정이 필요하다.

DIP는 디지털 아이템은 다운로드하고, IPMP(Intellectual Property Management & Protection) 정보를 처리하고, 저작권관련 정보를 핸들링하며, 디지털 아이템의 자원을 관리함과 동시에 DIM의 실행하거나 관리 한다. DIM은DIP의 한 부분이며, 디지털 아이템을 처리하는 여러 방법 중 하나이다.

그림9는 디지털 아이템 처리가 어떻게 MPEG-21 터미널에서 이용되는지를 나타낸다. DIM 엔진(DIME)은 터미널에서 중심적인 역할을 한다. DIM이 실행되기 위해서는 터미널의 여러 부분과 통신 가능해야 한다. 디지털 아이템의 전체적인 처리과정은 어플리케이션의 특성에 따른다. 디지털 아이템이 최종 사용자에게 전달되기까지 거쳐야 하는 여러 과정에서 다양한 구성원이 서로 다른 DIM을 사용하게 된다. 결국 DIM은 사용자가 디지털 아이템을 사용하는데 메뉴의 역할을 하게 된다.

2. 디지털 아이템 기본 동작 인터페이스 및 확장 동작 인터페이스

디지털 아이템 메소드는 DID를 의도하는 형식 또는 절차대로 처리하기 위해 DIP 엔진에 의해 수행되는 일련의 절차적인 동작들로 구성되며, DID 내에 포함되어 DID 요소들을 처리, 식별하는 방법들로 조직화되어 있어야 하며, 발생하는 이벤트에 대해 처리할 수 있어야 한다. 디지털 아이템 기본 동작 인터페이스는 DI 내의 요소들의 디스플레이, 리소스와 어플리케이션

DIM 에 사용되는 동작 함수 종류

DID 조작 함수	DID 특정 함수	RDD 관련 함수	확장 동작 함수
<ul style="list-style-type: none"> - CreateDIDLNode - GetDIDLNodeByID - Etc. 	<ul style="list-style-type: none"> - PresentChoice - SetSelection - Etc. 	<ul style="list-style-type: none"> - Print - Play - Etc. 	<ul style="list-style-type: none"> - TypeText - SearchingHighlightingText - Etc.

〈그림 2〉 디지털 아이템 기본 동작 인터페이스의 종류

션 실행, DI 전체 또는 일부 편집 등의 기능을 수행할 수 있는 기본 동작들을 의미한다. 디지털 아이템 기본 동작 인터페이스는 표준 규격 부분으로서 응용 프로그램 인터페이스(API: Application Program Interface)에 대한 신택스와 시멘틱스를 정의하고, 디지털 아이템 메소드 언어로는 ECMA스크립트 언어를 표준 언어로 규정 하였다. ECMA스크립트 언어를 이용하여 디지털 아이템의 처리 방식을 상호 호환적(표준화된) 형태로 프로그램화 할 수 있다. 이때 기본 동작 인터페이스(표준) 또는/그리고 확장 동작 인터페이스(비 표준) API를 사용하여 ECMA스크립트 언어로 표현된 프로그램을 디지털 아이템 메소드(DIM: Digital Item Method)라 하며 DID 문서 내에 포함되어 전달된다. 그림 2는 현재 디지털 아이템 처리에 대한 위원회 표준초안에 정의된 기본 동작 및 확장 동작 인터페이스들의 종류를 나타낸다. 기본 동작 인터페이스는 DID을 처리함에 있어 가장 기본이 되는 기능들을 담당하기 위해 트리 구조화된 디지털 아이템의 DID 노드를 조작할 수 있는 인터페이스인 DID 조작 기본 동작 인터페이스, DID 관련 노드 정보를 표현하는 DID 특정 기본 동작 인터페이스, RDD에 정의된 동사(verb)와 관련된 대응 동작 인터페이스를 규정하고 있는 RDD

관련 기본 동작 인터페이스, 그리고 비표준으로서 특정 응용 목적을 위해 정의된 확장동작(DIxO: Digital Item eXtension Operation) 인터페이스로 분류된다.

표1은 현재 디지털 아이템 처리에 대한 위원회 표준 초안에 채택된 디지털 아이템 기본 동작 인터페이스와 그 기능에 대해 나타낸다. 총 21개의 기본 동작 인터페이스들이 DID 조작 및 표현, 그리고 RDD 관련 동작을 정의하고 있다. 그러나 향후 표준화 과정에서 이러한 디지털 아이템 기본 동작 인터페이스들의 수는 필요한 요구 기능에 따라 증가하리라고 예상한다.

디지털 아이템 기본 동작 인터페이스의 사용에 대한 구체적인 설명을 위해 표 2는 기본 동작 인터페이스중의 하나인 GetDIDLNodeByID()에 대한 신택스 및 시멘틱스를 정의하고 있다. GetDIDLNodeByID() 인터페이스는 DID 문서가 트리로 표현 되었을 때 노드 아이디로 특정 노드에 대한 접근을 위해 사용된다.

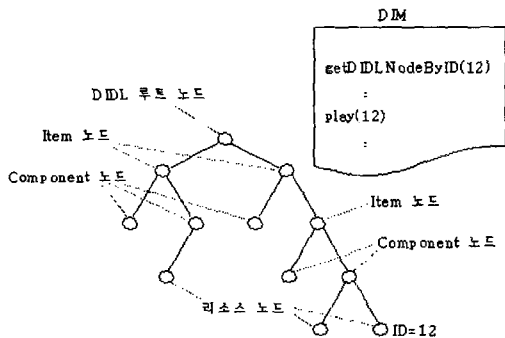
그림 3은 DID 데이터 파일이 파싱된 후 DOM 트리 형태로 된 논리적 구조를 나타낸다. 예시된 DIM 내에서 getDIDLNodeByIS(12)는, 기본 동작 인터페이스로서, 실행될 경우 "ID"가 12인 리소스 노드에 대한 주소를 리턴한다. Play(12) 기본 동작 인터페이스를 통해 "ID"가

〈표 1〉 디지털 아이템 기본 동작 인터페이스 및 기능

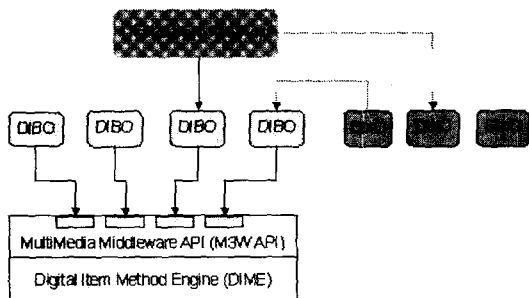
디지털 아이템 기본 동작 인터페이스	기능
StoreNewDID(didChanges)	새로운 DIDL 객체를 생성하여 didChanges 파라미터 객체에 규정된 변경 내용에 따라 DIDL 객체 변경하고 저장한다.
StoreDIDAs (sourceDID, didChanges)	sourceDID에 의해 지정된 또는 현재 DIDL 객체 에 didChanges 파라미터 객체에 규정된 변경 내용을 적용하여 저장한다.
StoreDIDOver (sourceDID, didChanges)	sourceDID에 의해 지정된 TH는 현재 DIDL 객체에 didChanges 파라미터 객체에 규정된 변경 내용을 적용하여 다시 덮어 저장한다.
LoadDID(locationURI)	locationURI에 의해 지정된 DID 데이터를 로드한다.
GetDIDLNode (docLocation, rootNode)	docLocation에 의해 지정된 위치의 노드에 대한 데이터를 배열 형태로 읽어온다.
GetDIDLNodeByID (sourceDID, id)	sourceDID에 의해 지정된 DIDL 객체에 대해 id 값을 가지는 노드에 대한 데이터를 배열 형태로 읽어온다.
GetDIDLNodeAttribute (DIDLNode, attribute)	DIDLNode에 의해 지정된 노드의 attribute라는 이름을 가지는 속성의 값을 읽어온다.
GetDIDLStatementText (statementNode)	statementNode에 의해 지정된 Statement 요소의 텍스트 값을 읽어온다.
ConfigureChoice(choiceNode)	choiceNode에 의해 지정된 Choice 요소를 MpegDIDNode 객체의 배열 형태로 읽어 온다.
SetSelection (selectionNode, state)	selectionNode에 의해 지정된 Statement 요소의 상태 값을 state 값 (true, false, or unresolved)에 따라 지정한다.
IsConditionsSatisfied (node)	node에 의해 지정된 condition 노드의 조건 값을 검사한다.
PlayResource (resourceNode, changes, async)	resourceNode에 의해 지정된 Resource 요소의 리소스를 랜더링 한다.
Pause(task)	task에 의해 지정된 랜더링 수행을 잠시 멈춘다.
Stop(task)	task에 의해 지정된 랜더링 수행을 중단한다.
Resume(task)	task에 의해 지정된 랜더링 수행을 다시 시작한다.
Release(task)	task에 의해 지정된 랜더링 수행을 멈춘 후 제어를 응용에 넘긴다.
GetObjectArguments (objectTypes, requestMessages)	objectTypes에 의해 정의된 객체 타입의 파라미터 값을 배열 형태로 읽어 온다.
GetExternalArgument (mimeType, requestMessage)	리소스의 URL 값을 문자열 형태로 읽어 온다.
GetData (dataTypes, requestMessages)	사용자로부터 입력을 dataTypes에 의해 규정된 형태(boolean, string, 또는 number)로 읽어 들인다.
Wait(interval)	주어진 interval 시간 만큼 기다린다.
Alert (message, messageType)	경고 메시지를 출력한다.

〈표 2〉 디지털 아이템 기본 동작 인터페이스의 한 신택스 및 시멘틱스 정의 예

신택스	GetDIDLNodeByID(sourceDID, id)
서술	id 매개변수로 주어진 값과 XML 스키마 데이터 타입 ID의 속성을 갖는 현재 DID로부터 존재하는 DIDL node를 뽑는다.
매개변수	sourceDID DIDL 노드를 검색할 소스 DID을 표현하는: MPEG DIDDocument 오브젝트 id 검색될 DIDL 노드를 위한 데이터 타입 ID 속성을 기술하는 문자열예를 들면 Item 요소는 id 속성의 값이 될 수 있고 Choice 요소의 choice_id 속성의 값이 될 수 있다.
반환값	MpegDIDNode 오브젝트 또는 노드가 없을때는 null이 반환된다.



〈그림 3〉 DID 파싱 후 디지털 아이템의 논리적 구성 및 DIM 내 기본 인터페이스들과의 관계



〈그림 4〉 MPEG-21 디지털 아이템 처리의 개관

12인 리소스에 대한 플레이를 수행한다.

그림 4와 같이 DIM은 DIM 엔진에 의해 멀티미디어 미들웨어에 구현된 기본 동작 인터페이스 및 확장 동작 인터페이스들을 호출하여 실행된다. 즉, MPEG-21 터미널의 어플리케이션이 DID 문서 내에 정의된 특정 리소스를 처리하거나, 또는 DID 문서의 특정 노드에 접근하여 정보를 읽어 오거나 저장할 때에 해당 DIM을 구현한 ECMA 스크립터를 해석하여 기본 또는 확장 동작 인터페이스를 호출하는 디지털 아이템 메소드 엔진을 실행시킨다.

3. 바인딩

가) 디지털 아이템 기본 동작 인터페이스의 ECMA스크립트 바인딩

디지털 아이템 메소드를 저작(authoring)하기 위해서는 디지털 아이템 메소드 언어인 ECMA 스크립트 언어를 이용하여 디지털 아이템 메소드를 프로그래밍 한다. 이때 디지털 아이템 메소드는 디지털 아이템 기본 동작 인터페이스를 포함하게 되는데 디지털 아이템 메소드가 DIM 엔진에 의해 실행될 때 디지털 아이템 기본 동

〈표 3〉 ECMA스크립트 바인딩을 위한 DIML 객체 메소드

DIML 객체 타입	메 소 드	
MpegDIPException	getDIPErrorCode()	
	getNumberOfTypes()	
	getTypeName(index)	
MpegDIPObjectMap	getNumberOfObjects(typeName)	
	getObject(typeName, index)	
	getObjects(typeName)	
	None	
MpegDIDDocument	None	
MpegDIDNode	getName()	
MpegDIPResourceStatus	getStatus()	
MpegDIPResourceChangeObject	addAdaptResourceChange(metadata)	
	addApplyResourceChangesChange (myResource, resourceChanges)	
	addApplyDIDLChangesChange (myMpegDIDNode, didlChanges)	
	addAddNewDIDLChildNodeChange (DIDLNodeType, didlChanges, location)	
	addAddCopyDIDLChildNodeChange (childSource, didlChanges, location, deep)	
	addModifyDIDLNodeAttributeChange (attribute, newValue)	
	addRemoveDIDLNodeChange(DIDLNode)	
	addAddDIDLNodeToObjectMapChange(objectType)	
	MpegDIPDIDLChangeObject	addAddNewDIDLChildNodeChange (DIDLNodeType, didlChanges, location)
		addAddCopyDIDLChildNodeChange (childSource, didlChanges, location, deep)
addModifyDIDLNodeAttributeChange (attribute, newValue)		
addRemoveDIDLNodeChange(DIDLNode)		
addAddDIDLNodeToObjectMapChange(objectType)		

작 인터페이스들이 표준화된 라이브러리 형태로 실행 단말 미들웨어에 존재하여야 하며 ECMA스크립트 언어 해석기에 의해 미들웨어로부터 호출이 가능하여야 한다. 이러한 실행 단말 환경을 제공하기 위해서는 디지털 아이템 기본 동작 인터페이스들이 표준화된 인터페이스 형태로 정의 되어야한다. 표1에서와 같이 디지털 아이템 기본 동작 인터페이스의 인터페이스에 대한 선택스가 ECMA스크립트에 의해 바

인딩 되는 규격을 의미 한다.

디지털 아이템 메소드는 디지털 아이템을 처리하기 위해 여러 가지의 DIML 객체를 표3에 서와 같이 정의하고 있다.

나) 디지털 아이템 기본 동작 인터페이스의 자바 바인딩

MPEG-21 디지털 아이템 처리 표준화 초기에 디지털 아이템 메소드를 저작(규정)하기 위한

〈표 1〉 디지털 아이템 기본 동작 인터페이스의 자바언어 바인딩

```

package org.iso.mpeg.mpeg21.mpegj.dibo;
import org.w3c.dom.*;
/* Java Binding interface for the GetDIDLNode DIBO */
public interface GetDIDLNode {
    /* @param docLocation, @param rootNode, @return node retrieved. */
    public MpegDIDNode getDIDLNode(String docLocation,
        MpegDIDNode rootNode; throws MpegDIPException;
}

```

언어인 DIML로서 ECMA스크립트 언어와 자바 언어가 결합 하였으나 ECMA스크립트 언어가 간단하고 응용을 위한 메모리 사용 요구가 작다는 이유를 들어 DIML로 결정되었다. 그러나 자바 언어에서도 최소한의 디지털 아이템 메소드를 처리 하게 하기 위해 디지털 아이템 기본 동작 인터페이스를 자바 언어에 바인딩 하는 방식만은 표준의 한 부분으로 포함하기로 결정하였다. 표 4는 디지털 아이템 기본 동작 인터페이스의 자바 언어 바인딩에 대한 한 예를 나타낸다.

다) MPEG-J 기반 디지털 아이템 확장 동작 인터페이스 호출

MPEG-J는 MPEG-4 시스템 규격에 채택된 기술로서 MPEG-4 시스템의 리소스를 제어하기 위한 자바 플랫폼을 제공한다. 이러한 개념을 확장하여 MPEG-21 디지털 아이템 처리에서도 자바 바인딩과 더불어 자바 클래스에 대한 패키지에 디지털 아이템 기본 동작 인터페이스 및 디지털 아이템 처리 객체를 포함시켜 MPEG-J를 이용하여서도 디지털 아이템 메소드를 처리 할 수 있도록 하였다.^[2]

IV. 디지털 아이템 처리를 이용한 요약 비디오 처리 단말 구현 예^[4]

그림 5는 DID 데이터 파일에는 실제 디지털 아이템의 위치와 설명 등이 서술되어 있고 그 디지털 아이템들을 언제나 일정한 형태로 처리 되도록 하는 디지털 아이템 메소드를 기술한다. 아래 그림 5는 본 실험에서 사용한, MPEG-21 터미널에 사용된, 실제 DID 데이터 파일의 한 예로 SOCCER란 디지털 아이템에 대한 간단한 설명(Descriptor)과 사용하려는 리소스(참조 위치를 포함한 이미지, 동영상)에 대하여 서술하고 있다. 하나의 Item 내부에 그 Item을 기술하기 위해 여러 세부 Item들(summary 등)이 존재하고 그 Item들에 작용할 수 있는 2가지 DIM(PlayOriginal, Treeview 등)이 DIM_LIST라는 Item에 표현되어 있다.

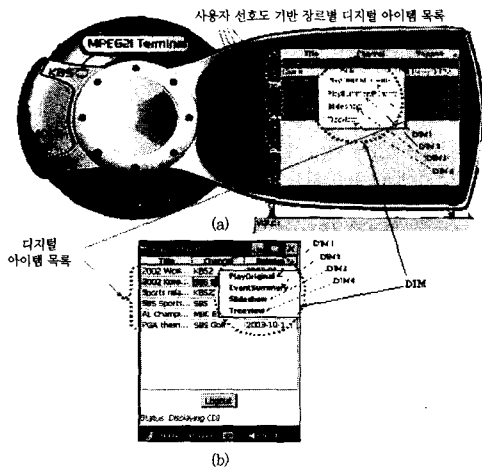
DID는 DID 파서를 통해 파싱되고 실제로 사용되는 디지털 아이템과 그 아이템에 작용하는 DIM이 어떤 것인지를 구분한다. 터미널 어플리케이션은 디지털 아이템에서 사용할 수 있는 DIM(PlayOriginal, Treeview)내에 DIBO(GetDIDLNode, Play 등)를 통하여 디지털 아이템 내 Descriptor에 서술되어 있는 텍스트 데이터나 아이템 리소스에 접근하여 그 실행 결과를


```

<Item id="DIM_LIST">
  <Component id="PlayOriginal">
    <Descriptor>
      <Statement mimeType="text/plain"> OriginalVideo</Statement>
    </Descriptor>
    <Resource mimeType="text/diml">
      function PlayOriginal (item){
        var Resource = GetDIDLNode ("Descriptor[4]/Component[1]", item);
        PlayResource (Resource)
      }]]&lt;/Resource&gt;
    &lt;/Component&gt;
  &lt;Component id="Treeview"&gt;
    &lt;Descriptor&gt;
      &lt;Statement mimeType="text/plain"&gt; OriginalVideo&lt;/Statement&gt;
    &lt;/Descriptor&gt;
    &lt;Resource mimeType="text/diml"&gt;<![CDATA[
      function Treeview(itemList){
        var summaryList = didDocument.objectMap.getObjects(GetNodeID(itemList)+ "SUMMARY_LIST");
        ShowTreeView(summaryList.length, GetDIDLNodeByID(GetNodeID(itemList)+ "SUMMARY"));
      }]]&lt;/Resource&gt;
    &lt;/Component&gt;
  &lt;/Item&gt;
&lt;Item id="SOCCER"&gt;
  &lt;Descriptor&gt;
    &lt;Component&gt;
      &lt;Resource mimeType="video/mpeg" ref="http://210.107.133.71/MPEG21TB/worldcup/soccer.mpg" /&gt;
    &lt;/Component&gt;
  &lt;/Descriptor&gt;
  &lt;Item id="SOCCER_SUMMARY"&gt;
    &lt;Item id="SSCENE1"&gt;
      &lt;Descriptor&gt;&lt;Statement mimeType="text/plain"&gt; Kick Off&lt;/Statement&gt;&lt;/Descriptor&gt;
      &lt;Descriptor&gt;
        &lt;Component&gt;
          &lt;Resource mimeType="image/jpeg" ref="http://210.107.133.71/MPEG21TB/images/pkickoff.jpg" /&gt;
        &lt;/Component&gt;
      &lt;/Descriptor&gt;
      &lt;Component&gt;
        &lt;Resource mimeType="video/wmv" ref="mms:// 210.107.133.71/MPEG21TB/worldcup/pkickoff.wmv" /&gt;
      &lt;/Component&gt;
    &lt;/Item&gt;
    &lt;Item id="SSCENE2"&gt;
      &lt;Descriptor&gt;&lt;Statement mimeType="text/plain"&gt;Shooting&lt;/Statement&gt;&lt;/Descriptor&gt;
      &lt;Descriptor&gt;
        &lt;Component&gt;
          &lt;Resource mimeType="image/jpeg" ref="http://210.107.133.71/MPEG21TB/images/pshooting.jpg" /&gt;
        &lt;/Component&gt;
      &lt;/Descriptor&gt;
      &lt;Component&gt;
        &lt;Resource mimeType="video/wmv" ref="mms://210.107.133.71/MPEG21TB/worldcup/pshooting.wmv" /&gt;
      &lt;/Component&gt;
    &lt;/Item&gt;
    .....
  &lt;/Item&gt;
&lt;/Item&gt;
</pre>
</div>
<div data-bbox="415 810 531 825" data-label="Caption">〈그림 5〉 DID 예제</div>
<div data-bbox="449 841 497 856" data-label="Page-Footer">-183-</div>
```

〈표 5〉 사용한 DIBO & DlxO

DIBO	DlxO
GetDIDLNode	ShowSlideShow
GetDIDLNodeByID	ShowSummaryVideo
GetNodeID	ShowTreeView
PlayResource	



〈그림 6〉 MPEG-21 터미널 GUI : a)PC, b)PDA

사용자 인터페이스에 나타낸다. 사용자가 선택한 DIM은 DIML 해석기에 의해 해석되고 DIM 엔진에 의해 기본 동작 인터페이스와 연결되어 기본 동작 인터페이스에 정의된 기능을 수행하게 된다. 실제적으로 비디오 요약물 트리뷰 형태로 프리젠테이션하는 트리뷰의 디스플레이는 확장 동작(Digital Item eXtension Operation: DlxO) 인터페이스에 의해 수행된다. 사용자가 선택한 DIM은 DIML 해석기에 의해 해석되고 DIM 엔진에 의해 기본 동작 인터페이스와 연결되어 기본 동작 인터페이스에 정의된 기능을 수행하게 된다. 실제적으로 비디오 요약물 트리뷰 형태로 프리젠테이션하는 트리뷰의 디스플레이는 확장 동작 인터페이스에 의해 수행된다.

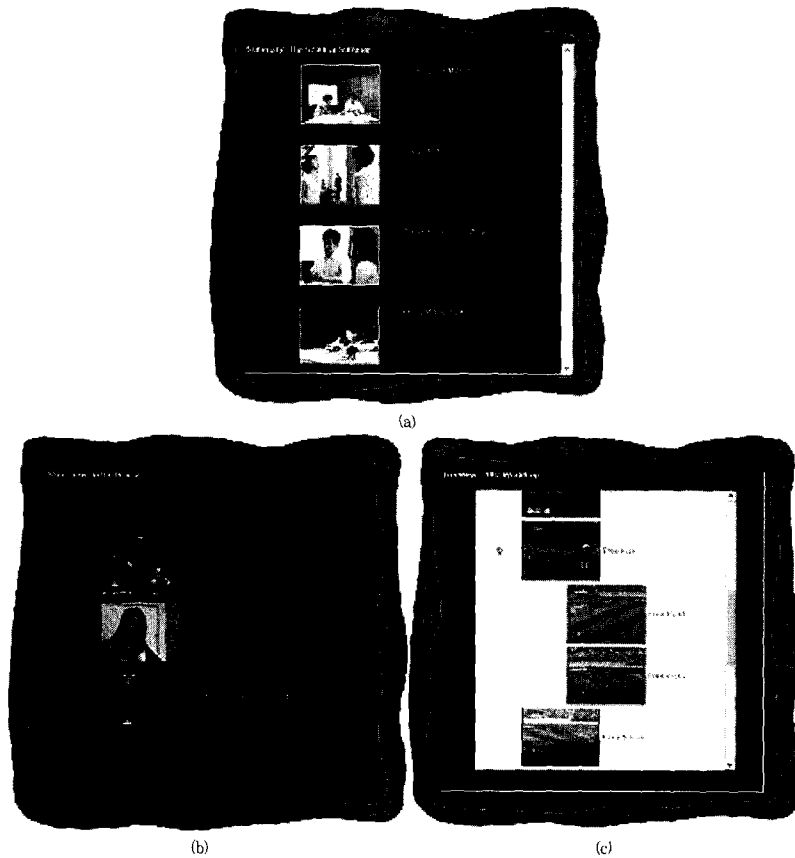
표 5는 MPEG-21 터미널의 DIM 엔진에서 사용한 DIBO와 DlxO의 목록이다.

그림 6은 PC와 PDA를 각각 플랫폼으로 하는 MPEG-21 터미널의 사용자 인터페이스를 나타내고 있다.

사용자가 로그인 하면 해당 사용자의 선호도가 포함된 MPEG-21 DIA 데이터를 DID 데이터로 표현하여 서버로 전송한다. 서버는 이에 응답하여 사용자의 선호도에 기반하여 제공 가능한 디지털 아이템에 대한 정보를 DID 데이터 형식으로 표현하여 MPEG-21 터미널에 전송한다. 터미널에 전송된 DID 데이터는 전송한 바와 같이 DID 파서를 통해 파싱되고 파싱된 DID 파일을 사용자 인터페이스를 통해 사용자의 장르 선호도에 따라 그룹핑하여 제시한다.

사용자의 선호도에 기반한 디지털 아이템 정보의 제시는 사용자로 하여금 원하는 정보에 쉽게 접근하는데 매우 유용하다. 그림 6-a)는 PC를 기반으로 하는 MPEG-21 터미널에서 사용자의 장르 선호도가 드라마, 오락, 스포츠 순이며 이에 따라 해당 디지털 아이템 정보가 그룹핑되었다. 이 밖의 장르는 기타(others)로 분류되어 그룹핑 되어있음을 보여준다. 한편, 그림 6-b)는 PDA를 기반으로 하는 MPEG-21 터미널에서 사용자의 장르 선호 순으로 디지털 아이템을 제시한다. 이 경우, 사용자 인터페이스의 물리적 공간의 제약으로 그림 6-a)의 경우처럼 개별 장르 별로 그룹핑 되어 별도로 제시되지는 않았다.

그림 7은 디지털 아이템에 포함된 DIM의 실행 결과를 나타낸다. 실험에서 사용된 디지털 아이템은 비디오 요약 서비스를 MPEG-21 터미널에 제공하는 것이며 제공되는 비디오 서비스 형식은 이벤트 기반 비디오 요약 보기, 요약 비디오의 슬라이드 쇼 보기, 트리뷰 보기 및 원 콘



〈그림 7〉 PC 기반 MPEG-21 터미널에서의 DIM 실행 결과 : a) 이벤트 기반 요약, b) 슬라이드 쇼, c) 트리뷰

텐츠 보기로 제공된다.

그림 7-a)는 이벤트 기반 비디오 요약 보기로서 한 드라마의 내용을 이벤트 별로 요약하여 각 이벤트 내용을 대표 프레임과 함께 텍스트로 표현 하였다. 그림 6의 예제에서 “EventSummary”라는 DIM을 선택하면 해당 DIM이 실행되어 그림 7-a)와 같이 디스플레이 된다. 이때 사용자가 특정 대표 프레임을 선택하면 해당 비디오 클립이 스트리밍 된다. 그림 7-b)는 대표 프레임으로 요약된 비디오 데이터를 슬라이드 쇼 형태로 나타내었으며, 그림 7-c)는 트리뷰에 대한 비디오 요약을 나타낸다. 서비스를 MPEG-21 터미널에 제공하는 것이며 제공되는 비디오 서비스

형식은 이벤트 기반 비디오 요약 보기, 요약 비디오의 슬라이드 쇼 보기, 트리뷰 보기 및 원 컨텐츠 보기로 제공된다.

V. 결론

본 논문에서는 MPEG-21 멀티미디어 프레임 워크에서 디지털 아이템의 처리 및 조작에 관한 상호 호환적 형태로 처리 방식을 규정하기 위한 표준으로서 MPEG-21의 파트 10인 디지털 아이템 처리에 관한 표준화 동향을 소개 하였다. 단 말에서의 디지털 아이템 처리 레벨에서의 규격화된 처리 방식을 표준화 하기위해 디지털 아이

템 기본 동작 인터페이스, 디지털 아이템 메소드 언어 및 관련 요소에 관한 표준화 작업을 진행하고 있다. 현재 디지털 아이템 처리 분야의 표준화를 주도하고 있는 호주의 Wollongong 대학과 덴마크의 Ghent 대학은 현재의 디지털 아이템에 대한 처리 범위를 DID 문서를 조작하고 약간의 RDD verb에 대응되는 기본 동작 인터페이스에 국한하여 표준화를 주도하고 있어 향후 본 규격이 완성되었을 경우 다양한 응용을 지원하기 위한 관점에서 보면 그 유용성을 그리 크지 않을 것으로 예상된다. 특히 이 분야의 표준화는 매우 제한된 기관에 의해 배타적 형태의 표준화 활동이 전개되고 있어 새로운 기술적 확장에 대한 표준화 작업이 어려울 것으로 전망된다. 이밖에도 미국의 Sun Microsystems 사, 한국방송공사, 한국정보통신대학교가 부분적으로 표준화에 참여하고 있다. 본 표준이 유용하게 사용되기 위해서는 다양한 응용을 지원할 수 있는 보다 확장된 디지털 아이템 기본 동작 인터페이스 세트와 이와 관련한 DIDL 규격 확장 등에 대한 추가 활동이 있어야 할 것으로 판단된다.

===== 참고문헌 =====

- [1] ISO/IEC JTC1/SC29/WG11, "Text of PDTR of ISO/IEC 21000-1 Second Edition," N6269, Hawaii, USA, December 2003.
- [2] ISO/IEC JTC1/SC29/WG11 "ISO/IEC 21000-10 CD Part 10 : Digital Item Processing," N6173, Hawaii, USA, December 2003.
- [3] ECMA Script Language Specification, ECMA 262, 3rd Ed., ECMA General Assembly, December 1999.
- [4] 손유미, 박성준, 김문철, 김종남, 박근수, "MPEG-21 터미널," 방송공학회 논문지, 2003년 제8권 제4호.

저자소개



김문철

1989년 경북대학교 전자공학과 공학사
 1992년 University of Florida, Electrical and Computer Engineering(석사)
 1996년 University of Florida, Electrical and Computer Engineering(박사)
 1997년-2001년 한국전자통신연구원, 선임연구원
 2001년-현재 한국정보통신대학교 공학부 조교수
 주관심분야 MPEG-4/7/21, 멀티미디어 정보처리 및 통신, 디지털 대화형 방송 미디어



김종남

1995년 금오공과대학교 전자공학과 공학사
 1997년 광주과학기술원 정보통신공학과 석사
 2001년 광주과학기술원 기전공학과 박사
 2001년-현재 KBS 기술연구소
 주관심분야 멀티미디어 데이터 압축, 멀티미디어 신호처리, 멀티미디어 통신, 워터마킹, MPEG-4/7/21



박근수

1982년 서울대학교 제어계측공학과 공학사
 1984년 한국과학기술원 전기전자공학과 석사
 1984년-현재 KBS 기술연구소
 주관심분야 HD/SD 방송장비 개발, 영상 처리, 워터마킹, MPEG-21