

# 온톨로지 기반의 모션 캡처 데이터베이스 설계 및 구현

정 현 숙<sup>\*</sup>

## 요 약

본 논문에서는 모션 캡처 데이터의 효과적인 저장 및 의미 기반 검색을 위한 프레임워크를 제안한다. 모션 캡처 기술은 현실감있는 캐릭터 동작을 얻기 위해 많이 사용되고 있지만, 모션 캡처 데이터의 검색과 저장을 위한 표준화의 부족으로 인한 여러 가지 문제점을 가지고 있다. 또한 미리 캡처된 모션 데이터에 의미론적 부가설명이 없으므로 애니메이터들이 캡처된 모션 데이터로부터 필요한 부분 동작들만을 검색하고 조합하여 새로운 동작을 생성하기가 어렵다. 본 논문의 목적은 모션 캡처 데이터의 재사용성을 향상시키기 위한 것이다. 먼저 상이한 모션 캡처 데이터 포맷들을 통합하기 위한 표준 포맷을 제안한다. 제안하는 표준 포맷은 XML 기반의 마크업 언어로서 MCML(Motion Capture Markup Language)라고 한다. 제안하는 MCML은 서로 상이한 포맷들의 변환 또는 통합을 하기 위해 유용할 뿐만 아니라, MCML 파일로 모션 데이터 베이스를 구축하므로 모션 캡처 데이터의 재사용성을 향상시킬 수 있다. 또한 모션 캡처 데이터의 부분 동작들에 의미를 부여하고 동작들 사이에 의미론적 연결을 위해 모션 온톨로지를 정의한다. 온톨로지 기반 데이터 접근으로 인해 부분 동작 및 그와 연관된 동작들을 검색하고 항해할 수 있다.

## Design & Implementation of a Motion Capture Database Based on Motion Ontologies

Hyun-Sook Chung<sup>\*</sup>

## ABSTRACT

A framework for semantic annotation of human motion sequences is proposed in this paper. Motion capture technology is widely used for manufacturing animation since it produces high quality character motion similar to the actual motion of the human body. However, motion capture has a significant weakness due to the lack of an industry wide standard for archiving and retrieving motion capture data. It is difficult for animators to retrieve the desired motion sequences from motion capture files as there is no semantic annotation on already captured motion data. Our goal is to improve the reusability of motion capture data. To archive our goal first, we propose a standard format for integrating different motion capture file formats. Our standard format is called MCML (Motion Capture Markup Language). It is a markup language based on XML (eXtensible Markup Language). The purpose of MCML is not only to facilitate the conversion or integration of different formats, but also to allow for greater reusability of motion capture data, through the construction of a motion database storing the MCML documents. Second, we define motion ontologies that are used to annotate and semantically organize human motion sequences. This ontology-based approach provides the means for discovering and exploiting the information and knowledge surrounding motion capture data.

**Key words:** Motion Capture File Format(모션 캡처 파일 포맷), Maekup Language(마크업 언어), Semantic Annotation(의미기반 동작표현), Ontologies(온톨로지)

※ 교신저자(Corresponding Author) : 정현숙, 주소: 경기도 부천시 원미구 역곡2동 산 43-1(420-743), 전화: 02)2164-4942, FAX: 02)2164-4777, E-mail: hsch@catholic.ac.kr

접수일 : 2005년 3월 15일, 완료일 : 2005년 4월 29일  
<sup>\*</sup> 정희원, 가톨릭대학교 컴퓨터정보공학부 교육전담교수

## 1. 서 론

모션 캡처 기법은 실세계의 캐릭터 움직임을 특수 카메라로 포착한 다음 이를 컴퓨터상의 가상 캐릭터에 적용하는 동작 생성 기법중 하나로서, 다른 동작 생성 기법과는 다르게 실제 세계의 캐릭터의 동작을 그대로 재현할 수 있다는데 큰 이점이 있다. 최근에는 동작을 캡처할 수 있는 장비의 향상으로 고품질의 동작 데이터를 획득할 수 있다.

그러나 모션 캡처 기술은 결정적인 단점을 갖고 있는데 현실감 있는 캐릭터 모션을 얻기 위해서는 기술력 있는 고수준의 인력, 고가의 장비, 많은 시간과 노력의 투자가 있어야 한다는 점이다. 또한 어렵게 생성한 고품질의 동작 데이터는 특정 캐릭터 및 상황에 맞춰진 고정화된 데이터로써 다른 목적으로 편집 또는 수정하여 재사용하기가 쉽지 않다는 것이다. 이러한 문제점을 극복하기 위해 많은 연구자들이 모션 합성 기술을 연구해오고 있다[5,8].

하지만 여러 동작들을 조합하여 새로운 동작을 생성하기 위해서는 모션 캡처 데이터가 가지는 다음의 몇 가지 문제를 먼저 해결해야 한다. 첫째, 모션캡처 데이터 포맷의 다양성 문제이다. 모션캡처 장비에 따라 서로 다른 구조의 데이터를 생성함으로써 다른 장비에 의해 캡처된 데이터의 통합, 교환, 공유 등을 어렵게 한다. 둘째, 이미 캡처된 데이터는 파일시스템에 의해 저장 및 검색되므로 접근성 및 데이터 중복의 문제가 발생한다. 현재 서비스되고 있는 모션 캡처 데이터 라이브러리를 보면 동일한 동작 데이터를 가지는 서로 다른 포맷의 파일들이 제공되고 있다. 셋째, 캡처된 데이터 파일에서 일련의 프레임들로 구성된 부분적인 동작 데이터를 검색하고 추출하기가 어렵다. 모션 캡처 데이터 라이브러리에서는 파일단위로 제공되며 파일의 동작 데이터에 대한 설명이 부족하다. 또한 하나의 파일에 들어있는 동작을 부분적으로 검색할 수 있는 기능이 없으므로 여러 파일들에서 원하는 동작 프레임들만 선택해서 추출하기가 어렵다.

본 논문에서는 이러한 문제를 해결하기 위해 먼저, 다양한 포맷의 모션 캡처 데이터를 통합하기 위한 XML(eXtensible Markup Language)[22] 기반의 표준 마크업 언어인 MCML(Motion Capture Markup

Language)[6]을 정의하였다. 서로 다른 포맷의 모션 캡처 데이터 파일을 하나의 단일화된 MCML 파일로 변환하여 저장함으로써 데이터의 중복성을 해결하였으며 XML 기반의 데이터 구조이므로 엘리먼트 수준의 검색이 가능함으로써 캡처된 데이터의 부분적 동작 검색 문제도 해결하였다. 또한 모션 캡처 데이터 라이브러리의 접근성 문제는 관계형 또는 XML 데이터베이스에 MCML 파일을 분해하여 저장함으로써 해결할 수 있다. 본 논문에서는 관계형 데이터베이스에 저장하기 위해 MCML의 엘리먼트를 테이블과 컬럼으로 매핑하는 논리 스키마를 설계하였다.

현재 모션 캡처 데이터 라이브러리에서는 카테고리를 탐색하거나 키워드 검색으로 모션 캡처 데이터 파일을 찾은 다음 파일 전체를 다운로드 받는 방식을 제공한다. 그러나 모션 합성을 위해서는 모션 클립(motion clip) 즉, 특정 위치의 시작 프레임에서 일련의 프레임들을 선택적으로 추출할 수 있어야 한다. 또한 여러 파일들에서 유사한 동작의 프레임들을 동시에 검색이 가능해야 한다. 이를 위해 본 논문에서는 하나의 MCML 파일의 전체 프레임들에서 부분적 모션 클립들을 지정할 수 있는 태그를 정의하였으며 모션 클립 각각의 동작을 설명하고 클립들의 동작 관계를 의미적으로 연결할 수 있는 모션 온톨로지를 정의할 수 있는 프레임워크를 제안하였다.

온톨로지는 특정 도메인이 가지는 지식의 주요 개념들을 표준 용어로 정의하고 이들을 계층적 및 의미적 연관성에 따라 연결함으로써 의미 기반 지식 검색이 가능하도록 하는 것으로 현재 시맨틱웹(Semantic Web), 지식관리시스템(KMS, Knowledge Management System), 이러닝(e-learning), 전자상거래(E-Commerce) 등 여러 분야에서 의미 기반 데이터 처리를 위한 핵심 도구로 사용되고 있다[13]. 온톨로지를 컴퓨터상에 구현하기 위한 데이터 모델로 RDF(Resource Description Framework)[3,17]와 토픽맵(Topic Maps)[6,9]이 있고 이 모델 상위에 온톨로지 추론을 지원하기 위해 로직을 기술할 수 있는 OWL 웹 온톨로지 언어[11]가 있다. 본 논문에서는 모션 온톨로지를 구현하기 위해 토픽맵을 사용한다. 토픽맵과 RDF는 서로 호환이 가능하며 많은 부분 비슷하지만 RDF가 URI로 도달 가능한 것들을 연결하는 자원(resource) 중심인데 반하여 토픽맵은 구체적인 사물이나 추상적인 개념 등 URI로 도달 가능하지 않

는 것들도 연결할 수 있는 주제(subject) 중심이기 때문이다[18].

본 논문에서는 여러 포맷의 모션 캡처 데이터 파일과 MCML 파일 사이의 상호변환과 MCML 파일의 관계형 데이터베이스로의 저장 및 모션 온톨로지 기반의 모션 클립 검색을 지원하는 시스템을 구현하였다. 이 시스템은 크게 모션 데이터 관리 모듈과 모션 온톨로지 관리 모듈로 구성된다. 모션 데이터 관리 모듈은 그 내부에 매핑관리자(mapping manager), 변환기(translator), 저장관리자(storage manager)를 가지고 모션 온톨로지 관리 모듈은 그 내부에 토픽맵 해석기(topic maps parser), 토픽맵 구축기(topic maps builder), 토픽맵 저장관리자(topic maps storage manager), 토픽맵 검색기(topic maps navigator)를 가진다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 관련 연구에 대해 소개하고, 3장에서는 MCML의 구조와 내용에 대해 자세히 설명한다. 4장에서는 의미 부여를 위한 모션 온톨로지의 구조와 데이터 모델을 소개한다. 그리고 5장에서는 MCML 파일 관리와 모션 온톨로지 생성을 위한 시스템 구현을 소개하고 6장에서는 실험을 통하여 구현된 시스템의 결과를 보인다. 마지막 7장에서는 본 논문의 결론 및 향후 연구방향을 기술한다.

## 2. 관련 연구

우선, 캐릭터 애니메이션, 가상현실 그리고 다른 분야에서 데이터를 표현하기 위한 방법으로 XML 기반의 새로운 마크업 언어 개발과 관련된 기존 연구를 개략적으로 요약할 수 있다. Morales[4]는 본 논문의 MCML과 유사한 XML기반의 모션 캡처 데이터 저장 방법을 고안했다. 이 방법에서 모션 캡처 데이터는 XML 데이터 형식으로 변환되어 저장됨으로서 애니메이션들이 웹 기반 환경 같은 상호 협력적인 환경에서 공동의 데이터에 접근 가능하게 하였다. 본 논문의 연구와 비교해 볼 때, 이 접근법은 세그먼트 단위의 간단한 형식으로 저장된 모션 캡처 데이터만을 다룰 수 있으며 캐릭터 구조체(skeleton)를 형성하는 계층적 구조는 고려되지 않았다. 더구나 본 논문의 MCML 언어와 같은 모션 캡처 데이터 통합을 위한 표준 마크업 언어의 구현을 제시하지 않았고 단지 XML을 사용한 데이터 변환의 가능성만을 언급했을

뿐이다.

VRML(Virtual Reality Modeling Language)[23]는 웹상에서 3차원 환경을 구현하기 위해 제작된 언어이고 H-ANIM[24]은 아바타의 3차원 모델링에 사용되는 구조를 설명하는 것으로 Web3D 협회에 의해 설립된 표준 규정이다. H-ANIM에서 휴머노이드(humanoid)의 명세화는 VRML 2.0에서 사용된 휴머노이드를 나타내는 표준 방법을 따르고 있다. H-ANIM의 휴머노이드 노드의 구조는 모션 캡처 데이터의 구조와 유사하다. 이는 조인트(Joint), 세그먼트(Segment), 사이트(Site), 뷰포인트(Viewpoint) 노드들로서 인체 형태의 특징을 나타내는 골격 및 결합 구조를 정의하는 전반적인 역할을 하기 때문이다. H-ANIM은 온라인상의 가상 환경에서 인간 캐릭터를 표현하는데 있어서 적합한 언어이긴 하지만 H-ANIM이 가지고 있는 너무 많은 부가적인 구문들로 인해 표준적인 모션 캡처 데이터 포맷으로 사용하기에는 너무 복잡한 언어이다.

모션 캡처 데이터의 프레임 단위가 아닌 연속된 프레임의 집합인 세그먼트를 분리하고 제약조건을 만족하는 세그먼트들을 합성하여 하나의 동작을 완성하는 연구로서 [5,8,15,16]이 있다. 이들 연구는 모션 데이터의 의미 정보를 활용하기 보다는 예제(example) 동작과 유사한 동작을 제약조건 기반으로 분석하여 탐색하는 방법이다. 부분적 모션 클립을 검색하는 것과 관련된 연구로서 비디오 데이터에서 특정 구간을 검색하기 위해 색감, 색상, 텍스처와 같은 저수준의 정보 또는 자유 형식의 텍스트 설명과 같은 고수준의 정보를 기반으로 하는 비디오 데이터 검색이 있다[10,20]. 이 비디오 데이터 검색의 자유 형식 텍스트 기술 방법은 비표준적이고 여러 비디오 데이터 사이에 공유가 어려운 점이 있으나 온톨로지에 기반하여 의미 정보를 부여할 경우 여러 모션 데이터의 동작에 대한 표준적인 의미 정의와 다양한 연관관계를 정의할 수 있으며 이러한 의미 정보는 모든 모션 데이터들 사이에 공유가 가능해진다.

## 3. MCML DTD 명세화

이 장에서는 MCML의 태그명과 문서 구조에 대해 설명한다. 현재 MCML은 ASF(Acclaim Skeleton File)/AMC(Acclaim Motion Capture data)[1], BVH(Biovision Hierarchical data)[2] 그리고 HTR(Hie-

rarchical Translation-Rotation)[12] 모션 캡처 데이터 포맷을 통합하기 위한 태그들을 정의하고 있다. 모션 캡처 데이터 포맷은 크게 아스키(ascii) 형식과 바이너리(binary) 형식으로 구분된다. 아스키 형식은 텍스트 편집기에서 파일의 내용을 볼 수 있고 쉽게 변형이나 추가 정보를 기록할 수 있다는 점에서 장점을 가지지만 파일의 크기가 크고 순차적으로 파일 내용을 읽어야 한다는 단점을 가진다. 바이너리 형식은 파일의 크기와 접근성에 있어서는 장점을 가지지만 내용을 쉽게 이해할 수 없고 형식에 맞춰진 프로그램에서만 읽혀진다는 단점을 가진다.

위의 ASF/AMC, BVH, HTR은 아스키 형식으로서 가장 보편적이며 산업 표준 모션 캡처 데이터 형식이며 대부분의 모션 소프트웨어에서 지원하고 있다[12,19]. 또한 현재 유료 또는 무료로 구할 수 있는 모션 캡처 데이터 파일의 대부분이 이 세 가지 파일 형식으로 제공되고 있다. 그러므로 MCML은 이 세 파일을 중심으로 통합할 수 있는 구조를 가진다. 그러나 XML의 특성에 따라 다른 포맷도 포함하도록 확장이 가능하며 바이너리 파일일지라도 전용 소프트웨어에서 아스키 파일로 변환이 가능하므로 이를 통해 MCML로 변환할 수 있다.

### 3.1 MCML 태그 정의

모션 캡처 데이터 형식을 살펴보면 크게 메타데이터에 해당하는 파일 헤더(header) 데이터와 캐릭터를 형성하는 캐릭터 구조체 데이터, 캐릭터의 동작을 나타내는 모션 데이터로 구성되어 있다. MCML 태그 집합은 이러한 유형별로 각각의 데이터 포맷에서

사용하는 예약어들에 대한 표준 용어로서 정의된다.

MCML의 헤더데이터 태그 정의. 모션 캡처 데이터 파일은 파일 유형이나 버전, 주식, 전체 프레임 수, 프레임 율 등과 같은 보조적인 정보를 포함하는 헤더데이터 영역을 갖고 있다. MCML은 헤더데이터를 표현하기 위한 태그를 다음 그림 1과 같이 정의하고 있다.

MCML의 캐릭터 구조체 태그 정의. 모션 캡처 데이터는 모션 장비로 캡처한 실세계의 사람을 모델링한 캐릭터의 구조체를 형성하는 계층적 데이터를 포함하고 있다. MCML은 캐릭터 구조체를 정의할 수 있도록 루트로부터 각 관절체까지의 거리, 관절체의 길이 및 관절체사이의 계층구조 등에 대한 태그를 정의하고 있다. MCML의 캐릭터 구조체 태그 집합은 ASF/AMC, BVH, HTR의 서로 다른 캐릭터 구조를 통합할 수 있도록 정의되어 있다.

그림 2는 MCML 파일에서의 캐릭터 구조체를 정의하는 일부분을 예로서 보이고 있다. root 엘리먼트는 캐릭터의 초기 위치를 가리키며, skeleton은 계층적 구조, 각 관절체의 위치 및 길이를 가리킨다. 계층적 구조는 관절체사이의 상하 포함관계로서 표현할 수 있다. name 엘리먼트는 각 관절체의 이름을 값으로 가지고 있는데, 각 관절체의 이름 또한 데이터 포맷마다 서로 다르게 사용되므로 MCML에서는 표준 관절체 이름을 아래와 같이 정의하였다.

MCML의 캐릭터 관절체명 정의. ASF/AMC, BVH, HTR 포맷은 캐릭터 구조체를 형성하기 위해 각기 다른 관절체 이름과 다른 계층적 구조를 정의하고 있다. 이들을 하나의 표준화된 구조체로 통합하기

|                        |         |                                |
|------------------------|---------|--------------------------------|
| filetype               | #PCDATA |                                |
| datatype               | #PCDATA |                                |
| filename               | #PCDATA |                                |
| version                | #PCDATA |                                |
| skeleton_name          | #PCDATA |                                |
| units                  | #EMPTY  | // mass, length, angle 속성 가진다. |
| num_segment            | #PCDATA |                                |
| num_frames             | #PCDATA |                                |
| dataframe_rate         | #PCDATA |                                |
| euler_rotation_order   | #PCDATA |                                |
| calibration_unit       | #PCDATA |                                |
| rotation_unit          | #PCDATA |                                |
| global_axis_of_gravity | #PCDATA |                                |
| bone_length_axis       | #PCDATA |                                |
| scale_factor           | #PCDATA |                                |

그림 1. MCML의 헤더데이터를 기술하기 위한 태그집합

```

<skeleton>
<root order="Tx Ty Tz Rz Rx Ry" axis="XYZ" position="0.00 0.00 0.00"
orientation="0.00 0.00 0.00"/>

  <bone id="1">
    <name> hips </name> <offset> 0.00 0.00 0.00 </offset>
    <channels num="6"> Tx Ty Tz Rz Rx Ry </channels>
    <bone id="2">
      <name> torso_1 </name><offset> 0.00 4.21 0.00 </offset>
      <channels num="3"> Rz Rx Ry </channels>
      <bone id="3">
        <name> neck1 </name><offset> 0.00 17.08 0.00 </offset>
        <channels num="3"> Rz Rx Ry </channels>
  
```

그림 2. MCML로 기술된 캐릭터 skeleton의 한 예

위해서 MCML에서는 이들보다 세분화된 관절체 이름을 정의하고 구조체 계층 구조를 구성하였다. 표 1은 MCML에 정의된 관절체 이름과 여기에 대응되는 각 데이터 포맷에서의 관절체 이름을 보여주고 있다.

BVH 파일 포맷은 인체 관절을 나타내기 위해서 마커(marker) 위치의 이름을 사용하는 반면 ASF/AMC 파일 포맷은 인체상의 뼈대 이름을 사용한다. HTR도 인체 뼈대의 이름을 사용하고 있지만 단지 ASF/AMC에 비해 관절의 수가 적다. MCML은 이러한 세 가지 포맷들을 모두 포함할 수 있을 뿐만 아니라 보다 상세한 구조체 정의를 위한 표현의 확장성을 가지고 있다.

MCML의 동작데이터 태그 정의. 동작 데이터는

전체 프레임 수, 한 프레임 당 플레이 시간, 한 프레임 당 움직임의 수, 한 프레임 당 로테이션의 수 등으로 구성되어 있으며 모션 장비로 캡처된 사람의 움직임을 그대로 재현하기 위한 값들을 가진다. MCML에서는 각 데이터 포맷에서의 동작 데이터들을 통합할 수 있는 표준 태그 집합을 아래 표 2와 같이 정의하였다.

### 3.2 MCML DTD 정의

MCML의 루트 엘리먼트인 <mcml>은 그 하위에 <meta>, <header>, <skeleton>, <motion> 엘리먼트들을 가진다.

META 엘리먼트. meta 엘리먼트는 8개 하위 엘리먼트를 가지고 있으며 MCML 문서 파일의 메타데이

표 1. MCML에 정의된 캐릭터 관절체 이름과 다른 포맷의 관절체 이름과의 매핑관계

|    | MCML            | ASF/AMC                    | BVH(1)     | BVH(2)         | HTR       |
|----|-----------------|----------------------------|------------|----------------|-----------|
| 1  | root            | h_root                     | root       | root           | undefined |
| 2  | head            | h_head(head)               | head       | head           | head      |
| 3  | neck1           | h_neck1(upperneck)         | neck       | neck           | undefined |
| 4  | neck2           | h_neck2                    | undefined  | undefined      | undefined |
| 5  | left_shoulder   | h_left_shoulder(lclavicle) | leftcollar | lshoulderjoint | undefined |
| 6  | left_up_arm     | h_left_up_arm(lhumerus)    | leftuparm  | lhumerus       | lupperarm |
| 7  | left_low_arm    | h_left_low_arm(lradius)    | leftlowarm | lradius        | llowarm   |
| 8  | left_wrist      | (lwrist)                   | undefined  | undefined      | undefined |
| 9  | left_hand       | h_left_hand(lhand)         | lefthand   | lwrist         | lhand     |
| :  | :               | :                          | :          | :              | :         |
| 50 | right_toe_two   | h_right_toe_two            | undefined  | undefined      | undefined |
| 51 | right_toe_three | h_right_toe_three          | undefined  | undefined      | undefined |
| 52 | right_toe_four  | h_right_toe_four           | undefined  | undefined      | undefined |
| 53 | right_toe_five  | h_right_toe_five           | undefined  | undefined      | undefined |

표 2. 동작 데이터를 기술하기 위한 MCML의 태그 집합과 다른 데이터 포맷의 예약어와의 매핑관계

|   | MCML                                     | ASF/AMC                               | BVH                                     | HTR  |
|---|--|---------------------------------------|---|--|
| 1 | motion                                   |                                       | motion                                  |  |
| 2 | frames                                   |                                       | frames                                  |  |
| 3 | frametime                                |                                       | frametime                               |  |
|   |  | frame                                 | frame                                   | frame                                      |
|   |  | Attribute:                            | Attribute:                              | Attribute:                                 |
| 4 | frame<br>Attribute:<br>id                | -<br>Tx<br>Ty<br>Tz<br>Rx<br>Ry<br>Rz | #Fr<br>Tx<br>Ty<br>Tz<br>Rx<br>Ry<br>Rz | frame#<br>Tx<br>Ty<br>Tz<br>Rx<br>Ry<br>Rz |
|   | frame_bone                               |                                       |   |  |
|   | Attribute:                               |                                       |   |  |
| 5 | name<br>Tx<br>Ty<br>Tz<br>Rx<br>Ry<br>Rz |                                       |   |  |

터를 기술한다. meta 엘리먼트의 구조는 그림 3에서 보이고 있다.

title 엘리먼트는 MCML 문서 파일에 주어진 제목이며 creator 엘리먼트는 모션 캡처 데이터 파일을 생성한 사람이나 기관을 말한다. subject 엘리먼트는 모션 캡처 데이터 파일의 캡처된 동작에 대한 주제(예: 발레, 춤 등)이며 description 엘리먼트는 동작에 대한 설명을 기술하는 곳이다. format 엘리먼트는 ASF/AMC, BVH, HTR 등의 모션 캡처 데이터 포맷

을 나타내며 duration 엘리먼트는 전체 프레임의 실행 시간을 나타낸다. category 엘리먼트는 동작의 카테고리(예: 스포츠, 이동, 인간의 상호작용 등)를 기술하는 곳이다. category, subject, title은 모션 캡처 데이터 파일을 카테고리별로 구성 및 검색하기 위해 사용된다.

HEADER 엘리먼트. header 엘리먼트는 15개의 하위 엘리먼트로 구성되어 있으며 그림 4에 그 구조를 보이고 있다. meta 엘리먼트와 달리 header 엘리

```

<! ELEMENT meta (title, creator, subject, description, date, format, duration, category)
<! ELEMENT title (#PCDATA)>
<! ELEMENT creator (#PCDATA)>
<! ELEMENT subject (#PCDATA)>
<! ELEMENT description (#PCDATA)>
<! ELEMENT date (#PCDATA)>
<! ELEMENT format (#PCDATA)>
<! ELEMENT duration (#PCDATA)>
<! ELEMENT category (#PCDATA)>
    
```

그림 3. MCML 메타데이터의 구조

```

<! ELEMENT header (filetype, datatype?, filename, version?, skeleton_name?, units?,
num_segments?, num_frames, dataframe_rate, euler_rotation_order?,
calibration_unit?, rotation_unit?, global_axis_of_gravity?,
bone_length_axis?, scale_factor?)
    
```

그림 4. MCML header 엘리먼트의 구조

리먼트는 MCML 파일의 데이터 특성을 기술하기 위해 사용된다.

**SKELETON** 엘리먼트. MCML의 *skeleton* 엘리먼트는 캐릭터의 계층적 구조를 정의한다. 그림 5에 *skeleton* 엘리먼트의 논리적인 구조를 보이고 있다.

*root* 엘리먼트는 캐릭터 계층 구조의 부모 엘리먼트로서 *axis*와 *order*속성은 초기 오프셋과 루트 노드 변환의 수행 순서를 나타낸다. *position* 속성은 루트 변환을 나타내며 *orientation* 속성은 을 정의한다.

*skeleton* 엘리먼트는 하나 또는 그 이상의 *bone* 엘리먼트를 갖고 있으며 *bone* 엘리먼트는 캐릭터의 계층 구조를 정의하기 위해 그 하위에 순환적으로 *bone* 엘리먼트를 가진다.

*name* 엘리먼트의 값으로는 표 1에 정의한 MCML 관절체 이름을 기술하여야 한다.

**MOTION** 엘리먼트. MCML *motion* 엘리먼트는 하나 이상의 *frame* 엘리먼트로 구성되어 있으며 *frame* 엘리먼트는 각 관절체의 프레임을 정의한다. *motion* 엘리먼트의 논리적인 구조는 그림 6에 나타나 있다.

*frametime* 엘리먼트가 각 프레임당 실행 시간인 반면 *frames* 엘리먼트는 프레임의 개수를 말한다. *frame* 엘리먼트는 *skeleton* 엘리먼트에서 정의된 각 관절체의 동작을 나타내기 위해 하나 또는 그 이상의

*frame\_bone* 엘리먼트를 가지며, 하나의 *frame\_bone* 엘리먼트는 한 프레임에서의 관절체의 전이 및 회전 값을 가진다.

#### 4. 모션 온톨로지 프레임워크 설계

모션 캡처 데이터에서 특정 프레임 구간의 모션 클립을 효과적으로 검색하기 위해서는 먼저 일정 프레임 구간을 나눈 다음 그 구간에 검색을 위한 동작 설명(motion description)을 기술해야 한다. 동작 데이터의 일정 구간을 나누고 그 위에 텍스트 설명을 부여하는 방법으로 그림 7과 같은 단편화(segmentation)과 층상화(stratification) 기법이 있다[10]. 이 방법들은 비디오 데이터에서 장면들을 나누고 각 장면에 텍스트 설명을 기술하는 것으로 비디오 검색을 위해 사용된다.

단편화 기법은 비교적 간단한 방법으로 연속적인 동작 데이터에서 시간의 흐름에 따라 겹치지 않도록 독립적인 일련의 시간 분할을 한 다음 각 단편(segment)에 동작을 묘사하는 구문을 기술하는 방법이다. 층상화 기법은 단편화 기법과 달리 여러 층(stratum)으로 겹쳐서 동작 분할을 한 다음 각각에 텍스트 설명을 기술하는 것이다. 이 기법은 포괄적인 서술에서부터 보다 상세한 서술까지 여러 수준에서

```

<! ELEMENT skeleton (root, bone+)>
<! ELEMENT root EMPTY>
<! ATTLIST root order CDATA #REQUIRED
              axis CDATA #REQUIRED
              position CDATA #REQUIRED
              orientation CDATA #REQUIRED>
<! ELEMENT bone (name, direction?, length?, position?, axis?, order?, dof?, limits?,
                 bodymass?, cofmass?, offset?, channels?, bone+)>
<! ATTLIST bone id ID #IMPLIED>
    
```

그림 5. MCML skeleton 엘리먼트의 구조

```

<! ELEMENT motion (frames, frametime, frame, motion_name*)>
<! ELEMENT frames (#PCDATA)>
<! ELEMENT frametime (#PCDATA)>
<! ELEMENT frame (frame_name, frame_bone+)>
<! ATTLIST frame id ID #IMPLIED>
<! ELEMENT frame_bone EMPTY>
<! ATTLIST frame_bone name CDATA #REQUIRED
                    Tx CDATA #IMPLIED   Ty CDATA #IMPLIED   Tz CDATA #IMPLIED
                    Rx CDATA #REQUIRED  Ry CDATA #REQUIRED  Rz CDATA #REQUIRED>
    
```

그림 6. MCML motion 엘리먼트의 구조

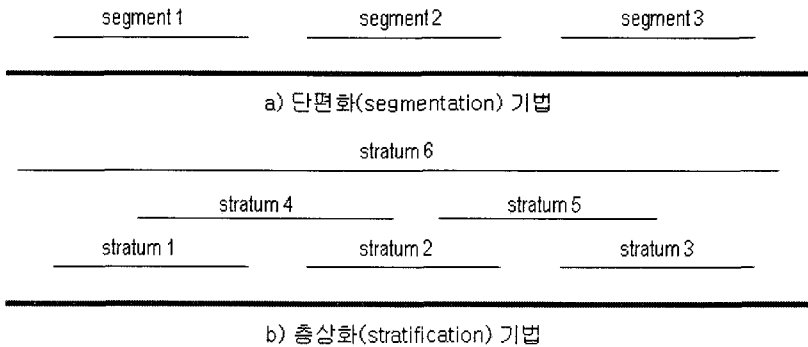


그림 7. 단편화 및 층상화 기법

동작 묘사를 할 수 있다는 장점을 가진다.

그러나 단편화와 층상화 기법은 동작에 대한 묘사와 동작 데이터를 분리하지 못하고 결합되어 있어서 유사한 동작을 가지는 여러 파일들에 중복적인 동작 묘사를 기술해야 하는 문제를 가진다. 이러한 문제를 해결하기 위해 본 논문에서는 모션 온톨로지 생성 프레임워크를 제안한다.

4.1 모션 온톨로지의 구조

모션 온톨로지는 동작을 묘사하는 용어(term)와 용어들 사이에 계층적 및 의미적 연관관계로 연결된 그래프 구조를 가진다. 그림 8은 모션 온톨로지의 개념적 구조를 보이고 있다. 모션 온톨로지는 지식층(knowledge layer)과 자원층(resource layer)으로 분

리되어 있으며 두 계층 사이에 연결 구조를 가진다. 지식층은 온톨로지들로 구성되어 있으며 상위 수준 온톨로지인 참조 온톨로지(reference ontology)와 동작의 주제별로 구축된 동작 기반 온톨로지들로 나누어진다. 동작 기반 온톨로지는 각 주제별로 여러 동작을 분류하고 각 동작을 묘사하는 표준 용어를 정의하고 있으며 동작 용어들 사이에 'is-a' 및 'part-of' 관계로 연결되는 계층적 구조와 의미적 연관성에 따라 연결되는 연관관계 구조를 가진다. 예를 들어, 그림 8의 댄스 온톨로지, 스포츠 온톨로지, 전투 온톨로지 등은 주제별로 구성된 온톨로지이고 상호작용 및 기본 동작 온톨로지는 일반적인 움직임용 서술하는 온톨로지이다.

동작 기반 온톨로지는 자원층의 실제 모션 클립으

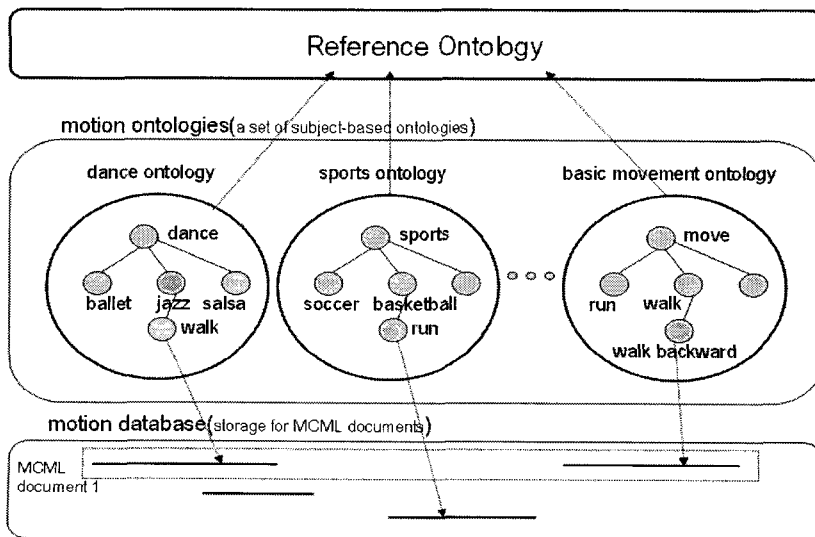


그림 8. 모션 온톨로지의 구조



로의 연결을 가진다. 예를 들어, 댄스 온톨로지의 dance → jazz → walk 계층 관계에서 walk 노드는 모션 데이터베이스의 댄스를 모션 캡처한 MCML 파일의 특정 프레임 구간의 모션 클립을 가리킨다. 지식층과 자원층의 분리로 인해 위에서 설명한 추상화 기법의 문제를 해결하며 지식의 표준화 및 공유를 제공한다. 서로 다른 동작 온톨로지들에서 동일한 모션 클립으로의 연결을 가질 수 있고 반대로 하나의 동작 노드가 여러 개의 모션 클립을 연결할 수 있다.

동작 기반 온톨로지의 상위에는 참조 온톨로지가 있다. 참조 온톨로지는 동작 기반 온톨로지에 대한 전역적인 용어 사전의 역할을 한다. 즉, 각 동작 기반 온톨로지의 공용 용어를 참조 온톨로지에 정의함으로써 중복 정의에 의한 문제를 해결하고 범용적인 용어를 공유할 수 있도록 한다. 또한 범용 용어들 사이의 유사어(synonym), 상위 개념어(hypernym), 하위 개념어(holonym) 관계를 제공함으로써 개념의 연결을 보장한다. 참조 온톨로지는 다양한 지식 도메인의 전문가들이 모여서 공용어를 정의하고 공용어의 의미적 연결 관계를 정의하여야 하지만 이는 매우 어려운 일이므로 기존의 일반 용어 사전을 사용할 수 있다. 이러한 일반 용어의 시소러스(thesaurus)로서 제공되는 것으로 워드넷(WordNet)이 있다. 워드넷에는 인간 움직임을 묘사하는 표준 용어를 정의하고 있으며 용어의 의미와 함께 개념적으로 상위어와 하위어 및 유사어 관계를 제공한다.

#### 4.2 동작 기반 온톨로지 구현

RDF/RDFS와 토픽맵은 온톨로지를 코드화하고 컴퓨터상에 구현하기 위해 사용하는 데이터 모델이다. 본 논문에서는 ISO 표준(ISO 13250)인 토픽맵을 사용하여 동작 기반 온톨로지를 구현할 수 있음을 보인다. RDF가 웹페이지의 하이퍼링크에 의미를 부여함으로써 웹상의 여러 가지 자원들을 의미적으로 연결하는 자원 중심의 구현인 반면 토픽맵은 웹상에 자원을 가지지 않는 추상적 개념 뿐만 아니라 웹페이지, 데이터베이스, 동영상 등 실제 자원들도 의미적으로 연결할 수 있는 주제 중심의 구현이다. 그러나 이 두 모델은 대부분 유사한 구조를 가지고 있으며 상호 변환이 가능하다.

토픽맵은 토픽(Topic), 연관관계(Association), 어커런스(Occurrence)를 기본 구성 요소로 하여 작성

된다[6]. 토픽은 토픽맵에 의하여 관리하고 탐색하고자 하는 대상(subject)을 컴퓨터가 이해할 수 있도록 표현된 정보 대상(information object)이다. 예를 들어, 스포츠 온톨로지의 동작 노드인 축구, 농구, 공을 차다, 공을 던지다 등이 토픽으로 정의된다. 어커런스는 토픽의 속성을 가리키는 것으로 동작명, 동작 범위, 동작 자원인 모션 클립의 주소, 동작 설명 등이다. 연관관계는 토픽들 사이의 의미적 연결로서 축구 토픽과 헤딩 슛 토픽 사이에는 상위-하위(superclass-subclass) 연관관계가 있으며 점프 토픽과 헤딩 슛 토픽 사이에는 의미적으로 'followed by'의 연관관계가 존재한다.

토픽맵을 기술하는 구문으로 XML 기반의 XTM(XML Topic Maps)[21]이 있다. 그림 9는 하나의 토픽과 하나의 연관관계를 정의하는 XTM 구문이다.

토픽의 id는 'walk001'이며 동작명이 'Walk backward'이며 dantce\_walk.mcml 파일의 frame 엘리먼트의 id가 'f000001' 부터 'f000214'인 구간을 자신의 모션 클립으로 가리키고 있다. 그리고 id가 'walk'을 상위 토픽으로 하는 superclass-subclass 계층적 연결을 가지고 있다. 모션 온톨로지를 구현하기 위해서는 XTM 구문에 적합한 온톨로지 문서를 작성한 다음 토픽맵 파서와 토픽맵 구축 모듈을 이용하여 메모리에 실제 토픽맵을 생성하거나 아니면 토픽맵 편집기 모듈을 이용하여 토픽들을 하나씩 추가함으로써 토픽맵을 직접 생성해 나갈 수 있다.

#### 5. 시스템 구현

구현된 시스템은 크게 모션 캡처 데이터 파일을 MCML 파일로 변환 및 저장하는 MCML 변환 관리 모듈과 모션 온톨로지의 생성 및 검색을 지원하는 온톨로지 관리 모듈로 나누어진다. MCML 변환 관리 모듈의 주요 기능은 모션 캡처 데이터 파일과 MCML 파일 사이의 상호 변환 기능, MCML 파일의 편집 기능, 모션 데이터의 플레이 기능, MCML 파일의 관계형 데이터베이스로의 저장 기능 등이며 온톨로지 관리 모듈의 주요 기능은 토픽맵 파싱 및 구축 기능, 토픽맵 편집 기능, 토픽맵 향해 및 검색 기능 등이다. 그림 10은 시스템의 전체적인 구조를 보이고 있다.

ASF/AMC, BVH 또는 HTR 포맷의 모션캡처 데이터 파일이 입력으로 들어오면 가장 먼저 MoCap

```

<topic id="walk001">
  <baseName><baseNameString>Walk backward<baseNameString></baseName>
  <occurrence>
    <instanceOf><topicRef xlink:href="#description"/></instanceOf>
    <resourceData>A person walking backward</resourceData>
  </occurrence>
  <occurrence>
    <instanceOf><topicRef xlink:href="#start-of-motion-clips"/></instanceOf>
    <resourceRef xlink:href="/resources/dance_walk_back.mcml#f000001"/>
  </occurrence>
  <occurrence>
    <instanceOf><topicRef xlink:href="#end-of-motion-clips"/></instanceOf>
    <resourceRef xlink:href="/resources/dance_walk_back.mcml#f000214"/>
  </occurrence>
</topic>
<association>
  <instanceOf><topicRef xlink:href="superclass-subclass"/></instanceOf>
  <member><roleSpec><topicRef xlink:href="superclass"/></roleSpec>
    <topicRef xlink:href="#walk"/></member>
  <member><roleSpec><topicRef xlink:href="subclass"/></roleSpec>
    <topicRef xlink:href="#walk001"/></member>
</association>

```

그림 9. 하나의 토픽과 하나의 연관관계를 정의하고 있는 XTM 구문

구문분석기(MoCap Syntax Analyzer)가 구문 분석을 시작한다. 이 구문분석기는 파일의 내용을 탐색하면서 유효한 토큰들을 추출하고 토큰테이블에 이 토큰과 값들을 저장한다. 토큰테이블에는 헤더 토큰, 스켈레톤 토큰, 그리고 모션 토큰들이 저장된다. 구문분석기에 의해 토큰테이블에 토큰들이 저장되면 MCML 변환기는 MCML 파일 생성에 이 토큰테이

블과 매핑테이블을 이용한다. 매핑테이블은 태그 매핑테이블과 관절체 매핑테이블로 구성된다. 관절체 매핑테이블은 MCML 관절체 명과 다양한 모션캡처 포맷의 관절체 명 사이의 매핑관계를 저장하는 테이블이고 태그 매핑테이블은 MCML 태그 집합과 다양한 모션캡처 포맷의 키워드들 사이의 매핑관계를 저장하는 테이블로서 헤더 데이터 태그집합, 캐릭터 구

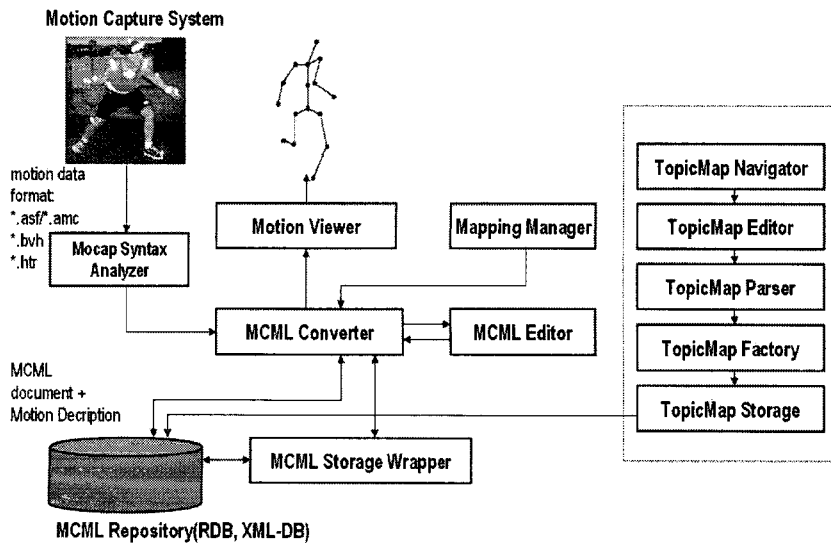


그림 10. 시스템 구조도 - MCML 관리 모듈 및 온톨로지 관리 모듈

조체 태그집합, 그리고 모션 데이터 태그집합을 내용으로 가진다. 이들 두 테이블은 MCML 변환기가 변환 과정에서 참조하는 사전으로 제공된다.

온톨로지 관리 모듈의 토픽맵 파서(TopicMap Parser)는 입력으로 들어온 XTM 문서의 구문을 해석한 다음 유효한 토큰들을 토픽맵 생성자(TopicMap Factory)에 전달한다. 토픽맵 생성자는 메모리내에 토픽맵을 생성하는데 해쉬 테이블에 그래프 구조로 토픽맵을 구현한다. 토픽맵 스토리지(TopicMap Storage)는 메모리의 토픽맵을 관계형 데이터베이스로 사상하는 기능을 수행한다. 토픽맵 편집기와 네비게이터는 사용자에게 웹 인터페이스를 제공하여 특정 키워드의 토픽들을 검색하거나 토픽들의 연결 구조를 따라 항해하면서 원하는 토픽을 찾도록 한다. 또한 토픽과 연결된 MCML 파일의 모션 클립들을 보여주고 모션 뷰어를 통하여 실제 동작을 보여준다. 또한 편집 기능을 이용하여 토픽의 추가, 수정 및 삭제가 가능하도록 한다.

## 6. 구현 결과

실험 데이터로 본 논문에서는 카네기 멜론 대학의 그래픽스 연구실에서 제공하는 모션 캡처 데이터 라이브러리의 파일들을 사용하였다. 이 파일들은 무료로 제공되고 있으며 댄스, 감정표현, 상호작용, 기본 동작 등의 카테고리별로 ASF/AMC, BVH, HTR 포맷의 파일들을 다운로드할 수 있도록 한다. 다운로드 받은 모션 캡처 데이터 파일은 MCML 변환기를 이

용하여 MCML 파일을 생성하고 아래 그림 11의 관계형 데이터베이스 스키마에 저장하거나 그림 12의 엑셀론(eXcelon) DXE Manager에 저장한다.

MCML 데이터베이스에서 모션 데이터를 검색하는 방법은 데이터베이스에 직접 접근하여 SQL 질의어나 XPath 질의어로 검색하는 방법과 모션 온톨로지를 이용한 지식 검색 방법이 있다. 모션 온톨로지를 이용한 지식 검색을 위해 온톨로지 내용에 대하여 전체 텍스트 색인(full-text index)을 구축하였으며 이 색인을 이용하여 키워드 검색을 지원하였다. 그림 13은 키워드 검색에 의해 모션 온톨로지서 검색된 댄스 동작 토픽들을 보인다.

검색된 동작 토픽을 선택시 해당 토픽의 상세 내용을 보이는 JSP 페이지를 실행한다. 상세 페이지에는 동작명, 동작 설명, 한글명, 상위 동작명, 연결된 모션 클립의 주소 등을 보인다. 모션 클립의 주소를 클릭하면 모션 데이터베이스에서 MCML 파일의 해당 프레임 구간을 검색하여 XML 구문으로 보이고 모션 뷰어를 이용하여 동작을 플레이한다.

본 논문에서는 모션 캡처 데이터 파일에서 모션 클립을 자동적으로 추출하는 자동 분류 문제는 다루고 있지 않다. 이 문제를 해결하기 위해 기존의 연구에서는 기계 학습 방법으로 학습 데이터를 미리 준비하여 분류 기계를 학습시킨 다음 입력된 모션 캡처 데이터가 어떤 학습 데이터와 근접한가를 계산하여 유사한 동작 데이터로 분류시킨다. 본 연구와 기계 학습 방법을 같이 적용할 경우 자동화된 모션 클립의 분류와 모션 데이터베이스 구축 및 모션 온톨로지에

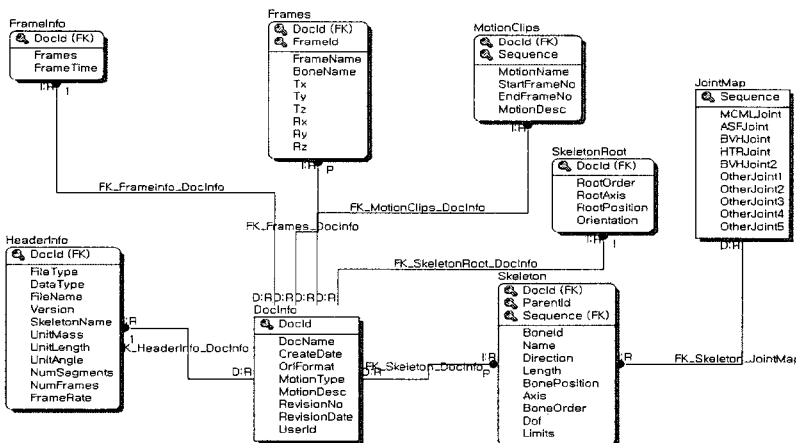


그림 11. MCML 파일을 관계형 데이터베이스에 저장하기 위한 스키마

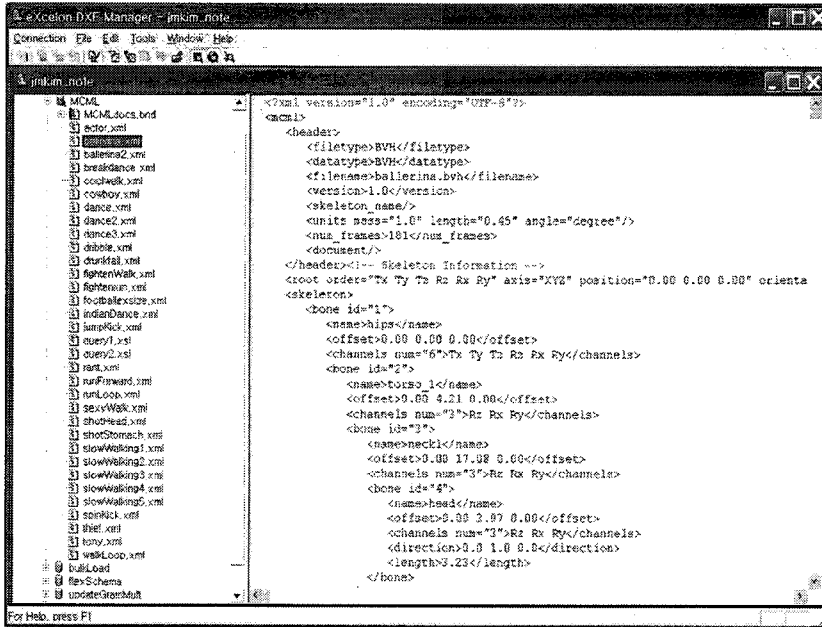


그림 12. eXcelon DXE Manager에서 저장된 MCML 파일을 검색한 화면

의한 의미 기반 검색을 지원할 수 있게 된다.

또한 본 논문에서는 모션 온톨로지 기반의 모션 클립 검색을 위한 프레임워크를 제안하고 있으며 실

제 표준적인 모션 온톨로지 자체를 제안하지는 않는다. 캐릭터 동작은 달리기, 점프, 회전 등 다양하고 광범위한 도메인을 가지기 때문에 표준적인 온톨로

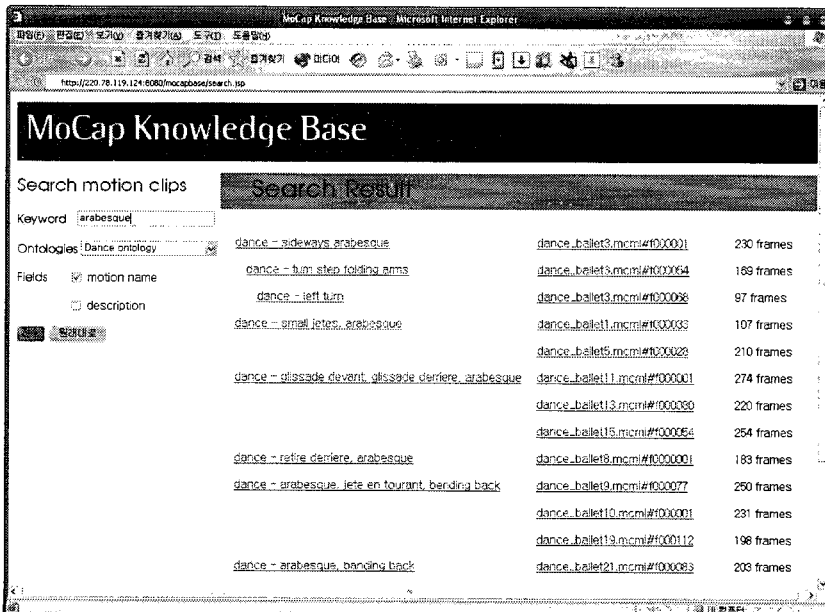


그림 13. arabesque 검색어로 검색한 결과 화면: arabesque 단어를 포함하는 동작명과 그 동작 토픽의 자원 파일인 MCML 파일의 프레임 시작위치와 프레임 길이를 검색 결과로서 보인다.

표 3. 기존의 모션 라이브러리와 모션 데이터베이스의 비교

| 평가항목     | 카네기멜론           | 마리오네트   | 모션 데이터베이스                         |
|----------|-----------------|---|-----------------------------------|
| 지원 포맷    | ASF/AMC,TVD,C3D | BVH,HTR,TRC, Maya, Light wave, SoftImage, 3D Studio | ASF/AMC,BVH,HTR                   |
| 표준(통합)포맷 | 없음              | 없음  | MCML                              |
| 검색및다운로드  | 파일              | 파일  | 파일, 모션클립, 캐릭터                     |
| 모션분류     | 디렉토리구조          | 디렉토리구조  | 모션 온톨로지                           |
| 키워드검색    | 주제번호, 모션설명      | 모션설명  | 온톨로지 토픽(개념)명                      |
| 질의어검색    | 불가능             | 불가능   | DB질의(XPath, SQL)<br>온톨로지질의(tolog) |
| 다중파일검색   | 불가능             | 불가능   | 다중파일에서 모션클립검색                     |
| 저장형태     | 파일시스템           | 파일시스템   | 데이터베이스(RDB,XMLDB)                 |
| 미리보기     | 동영상파일 제공        | 동영상파일 제공  | 모션뷰어에의한 플레이                       |

지를 구축하는 것은 매우 어려운 작업이다. 이는 다양한 분야의 전문가들이 모여서 서로 협의하에 동작을 묘사하는 표준 용어를 정의하고 용어들 사이에 계층적 및 연관적 의미 관계를 설정하여야 하는 많은 시간과 노력을 요구하는 작업이다.

예를 들어, 댄스 도메인에서도 발레, 재즈, 동양 무용 등 다양한 도메인이 있으며 각 도메인의 전문가들이 각자의 표준 온톨로지를 구축해야하고 나아가 유사한 도메인 온톨로지 사이에 공용 용어에 대한 표준화 작업이 이루어져야 한다.

표 3은 카네기멜론 대학의 모션 라이브러리(<http://mocap.cs.cmu.edu>), 한국첨단게임산업협회에서 제공하는 마리오네트 모션 라이브러리(<http://218.145.57.154/>) 및 본 논문의 모션 데이터베이스와의 기능을 몇 가지 항목으로 비교한 것이다. 이 표에서 보여지는 모션 데이터베이스의 성능 우위로 기존의 파일 단위 라이브러리에 비해 모션 데이터베이스가 모션 클립의 재사용을 높이는 데 더 유리함을 알 수 있다.

## 7. 결 론

본 논문에서는 다양한 모션 캡처 데이터 포맷을 통합하기 위한 XML 기반의 마크업 언어를 설계하였고 또한 의미 기반 모션 클립 검색을 위해 모션 온톨로지 적용을 위한 프레임워크를 제안하였다. 기존의 모션 캡처 데이터 라이브러리는 모션 캡처 데이터를 파일 단위로 저장함으로써 데이터의 중복, 공유의 어려움, 보안 및 관리의 어려움 등이 있다. 그러나 MCML로 변환되어 저장할 경우 상용 데이터베이스

를 활용함으로써 데이터의 공유, 트랜잭션 처리, 보안 등 여러 가지 문제를 해결할 수 있다. 또한 파일 전체를 검색하고 다운로드하는 대신 엘리먼트 수준의 작은 의미단위로 검색할 수 있으며 캐릭터 구조체와 모션을 분리하여 검색할 수도 있다.

모션 온톨로지는 모션 데이터베이스 상위에 의미 기반 검색을 위한 지식층을 제공함으로써 유사한 동작을 가지는 모션 클립들을 쉽게 검색할 수 있도록 한다. 본 논문에서는 토픽맵 기반의 모션 온톨로지 구축을 위한 시스템을 구현하였다. 토픽맵은 주제 중심의 개념 형식화를 제공하기 때문에 다양한 동작을 묘사하기에 적합하며 여러 주제의 동작 기반 온톨로지를 확장시키는 데 용이하다.

향후 연구 과제인 모션 클립의 자동 추출 및 온톨로지로의 자동 분류가 가능할 경우 본 논문의 모션 캡처 데이터베이스와 온톨로지는 게임 및 영화 제작, 스토리텔링 시스템, 캐릭터 애니메이션 등 모션 캡처 데이터를 활용하는 모든 분야에 지식 저장소로서 활용될 수 있으리라 본다.

## 참 고 문 헌

- [1] Acclaim, ASF/AMC File Specifications page, <http://www.darwin3d.com/game/dev/acclaim.zip>.
- [2] Biovision, BVH Specifications page, <http://www.biovision.com/bvh.htm>.
- [3] Brickley, D. and Guha, R.V., Resource Description Framework (RDF) Schema Specifica-

- tion. World Wide Web Consortium:[http://www.w3.org/TR/ PR-rdf-schema](http://www.w3.org/TR/PR-rdf-schema).
- [ 4 ] C.R. Morales, "Development of an XML Web Based Motion Capture Data Warehousing and Translation System for Collaborative Animation Projects," In Proceedings of the 9-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2001.
- [ 5 ] E. Hsu, S. Gentry, and J. Popovic, "Example-Based Control of Human Motion," Eurographics/ACM SIGGRAPH Symposium on Computer Animation, 2004.
- [ 6 ] Graham M.. "Topic Map technology—the state of the art," XML 2000 Conference & Exposition, Washington, USA, December 2000.
- [ 7 ] H.S. Chung and Y.B. Lee, "MCML:Motion Capture Markup Language for Integration of Heterogeneous Motion Capture Data," Computer Standards & Interface, Vol. 26, pp.113-130, 2004.
- [ 8 ] L.M. Tanco and A. Hilton, "Realistic Syntehsis of Novel Human Movements from a Database of Motion," In Proceedings of the IEEE Workshop on Human Motion HUMO 2000.
- [ 9 ] K. Ahmed, "Developing a Topic Map Programming Model," In Knowledge Technologies 2001.
- [10] M. Hacid, C. Declair, and J. Kouloumdjian, "A Database Approach for Modeling and Querying Video Data," IEEE Tranaction on Knowledge and Data Engineering, Vol. 12, No. 5, pp.729-750, 2000.
- [11] M. K. Smith, C. Weltry, and D. L. McGuinness, "OWL web ontology language guide," Technical report, W3C, <http://www.w3c.org/TR/2004/REC-owl-guide-20040210/>, 2004.
- [12] M. Meredith and S.Maddock, "Motion capture file formats explained," Department of Computer Science Technical Report CS-01-11.
- [13] M. Uschold and M. Gruninger, "Ontologies: Principles, Methods and Applications," Knowledge Engineering Review Vol. 11, No. 2, 1996.
- [14] Michel B., Martin B. and Steve N. ISO/IEC 13250 TopicMaps.
- [15] O. Arikan and D.A. Forsyth, "Interactive Motion Generation from Examples," In Proceedings of ACM SIGGRAPH 2002.
- [16] O. Arikan, D.A. Forsyth, and J.F. O'Brien, "Motion Synthesis from Annotations," In Proceedings of ACM SIGGRAPH 2003.
- [17] O. Lssila and R. Swick, "Resource description framework(RDF) model and syntax specification," W3C, 1999.
- [18] P. Ciancarini, M. Pirruccio, F. Vitali, R. Gentilucci, and V. Presutti, "Metadata on the Web. On the integration of RDF and Topic Maps," In Proceedings of Extreme Markup Languages 2003.
- [19] S. Rosenthal, B. Bodenheimer, C. Rose, and J. Pella, "The process of motion capture: Dealing with the data," In Proceedings of the 8th Eurographics Workshop on Animation and Simulation 1997.
- [20] S.P. Chao, C.Y. Chiu, S.N. Yang, T.G. Lin, and Tai Chi, "Synthesizer: a motion synthesis framework based on key-postures and motion instructions," Comp. Anim. Virtual Worlds 2004, vol. 15, pp.259-268, 2004.
- [21] TopicMaps.org Authoring Group, XML Topic Maps(XTM) 1.0, <http://www.topicmaps.org/xtm/1.0/>.
- [22] W3C, Extensible Markup Language (XML) 1.0, <http://www.w3c.org/TR/1998/RECxml-19980210/>, 1998.
- [23] Web3D Consortium, VRML International Standard, [http://www.web3d.org/technicalinfo/specifications/ISO\\_IEC\\_14772-All/index.html](http://www.web3d.org/technicalinfo/specifications/ISO_IEC_14772-All/index.html).
- [24] Web3D Consortium, H-ANIM 2001 Specification, <http://www.h-anim.or>



정 현 숙

1993년 2월 대구가톨릭대학교  
물리학과(이학사)

1995년 2월 대구가톨릭대학교  
전산학과(이석사)

2003년 8월 연세대학교 컴퓨터  
과학과(공학박사)

2004년 9월~현재 가톨릭대학교  
컴퓨터정보공학부 교육전담교수

관심분야: 캐릭터모델링, 모션그래픽스, 이러닝, 시맨틱웹