

# 모션 캡처 데이터 통합을 위한 새로운 마크업 언어의 구현

## (Implementation of New Markup Language for Integrating of Motion Capture Data formats)

정 현 속<sup>†</sup>      이 일 병<sup>\*\*</sup>  
(Chung Hyun Sook)      (Lee Il Byung)

**요 약** 최근 영화나 게임 산업에서 모션 캡처 기법은 현실적인 동작을 생성하기 위한 기술로서 널리 사용되고 있으며 이로 인해 공동 작업을 하는 여러 애니메이터들 사이에 모션 데이터를 공유하거나 서로 교환하는 일이 요구된다. 그러나 모션 캡처 데이터는 사용하는 모션 캡처 시스템에 따라 데이터 포맷에 많은 차이가 있다. 데이터 포맷이 상이함으로 인해서 저장된 모션 캡처 데이터의 재사용과 상호 교환에 많은 어려움이 발생된다.

본 논문에서는 이를 해결하기 위한 방법으로 모션 캡처 데이터를 통합할 수 있는 표준 포맷을 정의하고 이 표준 포맷을 이용한 모션 데이터 관리 시스템을 제안한다. 제안하는 표준 포맷은 XML(eXtensible Markup Language) 기반의 마크업 언어로서 MCML(Motion Capture Markup Language)라고 한다. 모션 캡처 데이터 관리 시스템은 모션 캡처 데이터를 이 MCML로 변환한 다음 데이터베이스에 저장하며 XML의 엘리먼트 단위로 검색할 수 있는 구조를 제공한다.

**키워드** : 모션 캡처 데이터, 모션 캡처 기술

**Abstract** Motion capture technology is widely used to make a realistic motion in these days. So, motion capture data is required to exchange between many animators in a work group. But different motion capture devices have different motion capture data formats. Thus it is difficult that a animator reuse and exchange motion capture data in a storage.

In this paper, we proposes a standard format for integrating a different motion capture data format. In addition, we proposes a framework of a system that manage motion capture data using our standard format. Our standard format is called MCML(Motion Capture Markup Language). It is a markup language for motion capture data and based on XML(eXtensible Markup Language). Our system to manage motion capture data consists of a several component - MCML Converter, MCML Editor, Motion Viewer.

**Key words** : Motion Capture Data, MCML(Motion Capture Markup Language), Motion Capture Technology

## 1. 서론

### 1.1 연구동기

최근 컴퓨터 애니메이션 분야에서 인체와 관련된 연구로서 캐릭터 애니메이션에서부터 게임제작까지 자연

스런 움직임을 표현하기 위하여 모션 캡처 기법을 사용하는 사례가 많이 있다. 컴퓨터 그래픽 애니메이션 기술은 제작 기술 방식에 따라 키프레임(Keyframing) 애니메이션, 물리법칙과 일련의 규칙들을 기반으로 하는 시뮬레이션 애니메이션, 모션 캡처(Motion capture) 애니메이션으로 분류할 수 있다[1]. 이 중에서 모션 캡처 기술은 모션 데이터를 생성하기 쉽고 물리적으로 완벽한 움직임을 생성하기 때문에 실시간 애니메이션의 문제를 해결하기 위해 많이 사용되는 방법이다[2].

모션 캡처 기법은 실세계의 캐릭터의 움직임을 특수

<sup>†</sup> 정 현 속 : 안양대학교 디지털미디어학부 교수  
hsch@csai.yonsei.ac.kr

<sup>\*\*</sup> 종신희원 : 연세대학교 컴퓨터과학과 교수  
yblee@csai.yonsei.ac.kr

논문접수 : 2002년 8월 28일

심사완료 : 2002년 11월 27일

카메라로 포착하여 컴퓨터 상의 가상 캐릭터에 적용하는 동작 생성 기법중 하나로서, 다른 동작 생성 기법과는 다르게 실제 세계의 캐릭터의 동작을 그대로 재현할 수 있다는데 큰 이점이 있다. 최근에는 동작 캡처할 수 있는 장비의 향상으로 고품질의 동작 데이터를 획득할 수 있다. 그러나 동작 캡처 기법으로 생성된 고품질의 동작 데이터는 특정 캐릭터 및 상황에 맞춰진 고형화된 데이터로써 다른 목적으로 편집 또는 수정하여 재사용하기가 그리 쉽지 않다.

특히 캡처된 모션 데이터는 모션 캡처 시스템에 따라 다른 데이터 포맷을 사용한다. 세그먼트 형태의 간단한 포맷에서부터 계층구조 형태의 복잡한 포맷까지 각 캡처 시스템들은 그 마다 캡처한 내용을 잘 표현할 수 있는 데이터 포맷을 나름대로 정의하고 있다. 문제는 서로 다른 모든 포맷을 이해하는 모션 소프트웨어를 개발하거나 기존 소프트웨어에 플러그인을 추가하는 것은 시간과 비용이 많이 드는 일이다[3].

또한 기존의 시스템에서는 모션 캡처된 데이터를 파일 단위로 라이브러리 형태로 제공한다. 물론 데이터베이스에 저장할 수도 있지만 이 역시 파일 단위로 저장된다. 그러나 애니메이터가 캡처된 모션 데이터를 재사용하기 위해서는 먼저 자신이 원하는 모션(예. 걷는다, 뛰다, 춤춘다 등)을 쉽게 검색할 수 있어야 하며 전체 모션의 일부 프레임들만 선택해서 검색해야 하는 경우도 있다. 이를 위해서는 모션 캡처 파일을 의미있는 정보 단위로 분리해서 저장해 두어야 하며 이 분리된 각 부분에 대해 검색할 수 있어야 한다.

그리고 컴퓨터 그래픽 애니메이션을 제작하거나 게임을 제작할 경우에 여러 애니메이터들이 한 캐릭터의 여러 모션을 나누어서 작업을 하는 경우가 많다[4,5]. 이 경우 모션 데이터를 상호간에 교환해야 하는데 서로 다른 포맷을 사용할 경우 각각의 플러그인이 없다면 교환이 어려워진다. 그리고 여러 모션 파일들에서 일부분의 모션 데이터를 합성해서 새로운 모션을 생성하는 것도 불가능하다[6].

본 논문에서는 서로 다른 포맷의 모션 캡처 데이터의 통합을 위한 표준 포맷을 정의하고 캡처한 모션 파일들이 표준 포맷으로 변환한 다음 파일 단위가 아닌 의미 있는 정보 단위로 분할하여 데이터베이스에 저장함으로써 모션 캡처 데이터의 재사용과 교환이 용이하도록 하였다.

## 1.2 연구범위 및 논문의 구성

먼저 서로 다른 포맷의 모션 캡처 데이터에 대한 표준 포맷을 정해야 한다. 이를 위해 본 논문에서는 XML

(eXtensible Markup Language)를 사용한다. XML은 서로 다른 포맷의 데이터를 통합하거나 교환하기 위한 구조화된 마크업 언어로서 여러 분야에서 데이터 표현 도구로 사용되고 있다.

애니메이션이나 가상현실 등에서도 XML을 이용하여 데이터를 표현하는 연구를 수행하고 있다. 본 논문에서는 모션 캡처 데이터를 표현할 수 있는 마크업 언어를 XML을 이용하여 제안하였으며 이를 MCML(Motion Capture Markup Language)라고 부른다. 이 마크업 언어는 최근 많은 모션 소프트웨어에서 지원하는 ASF/AMC, BVH, HTR 포맷을 통합할 수 있는 태그 집합을 정의하고 있으며 새로운 캡처 데이터 포맷도 쉽게 추가할 수 있는 확장성 있는 구조를 가진다.

MCML로 변환하여 데이터베이스에 저장할 경우 문서 전체를 검색할 수도 있지만 한 문서의 엘리먼트 단위로 세부적인 검색이 가능하다. 본 논문에서는 XML 전용 데이터베이스를 사용한다. XML 전용 데이터베이스는 XML 문서를 객체로 변환해서 저장하며 XQL, XPATH 등 XML 검색 언어를 지원한다. 모션 캡처 데이터 파일을 MCML 문서로 변환한 다음 이 데이터베이스에 간략한 설명과 함께 저장한 다음 특정 모션에 대해 검색할 경우 계층 구조로 내용을 검색할 수 있다.

검색된 MCML 문서는 MCML 편집기에서 캐릭터 관절체 값이나 모션의 프레임 값들을 직접 변경할 수 있다. 일반적인 모션 캡처 데이터 파일은 텍스트 형태로 제공되기 때문에 쉽게 읽고 변경하기가 어렵다. 그러나 MCML로 변환된 문서는 트리구조를 가지기 때문에 캐릭터의 특정 관절이나 모션의 특정 프레임을 쉽게 확인하고 변경할 수 있다.

이와 함께 여러 MCML 문서들에서 원하는 모션의 일부 프레임들을 모아서 새로운 동작의 MCML 문서를 생성할 수 있고 이 문서를 ASF/AMC, BVH, HTR 등으로 재 변환한 다음 모션 편집 소프트웨어에서 모션을 재생할 수 있다. 즉, 기존의 저장된 모션을 합성하여 새로운 동작을 생성할 수 있는 것이다. 그러므로 비슷한 구조의 캐릭터들의 같은 동작에 대해 별도로 모션 캡처를 할 필요가 없다.

본 논문의 구성은 다음과 같다. 1장에서는 모션 캡처 데이터의 표준 포맷의 필요성과 연구범위를 살펴보았고, 2장에서는 본 연구의 관련연구에 대해 소개한다. 3장에서는 모션 캡처 데이터의 구조와 내용에 대해 살펴보고, 4장에서는 모션 캡처 데이터의 표준 포맷으로 MCML을 소개한다. 그리고 5장에서는 MCML 기반의 모션 캡처 데이터 관리 시스템의 주요 요소의 설계 및 구현에

대해 설명한다. 마지막으로 6장에서는 결론 및 향후 연구 방향을 기술한다.

## 2. 관련연구

이번 장에서는 캐릭터 애니메이션, 가상현실 등의 분야에서 데이터를 표현하는 방법으로 XML을 이용하여 새로운 마크업 언어를 개발하는 연구가 진행되어 오고 있는데 이들 관련 연구를 살펴보고자 한다.

첫째로, 본 논문의 연구와 직접적으로 관련된 연구로서 XML에 기반한 모션 캡처 데이터의 저장[7]에 대한 논문을 들 수 있다. 이 논문에서는 상호협조적인 애니메이션 환경, 즉 웹 기반 환경에서 전체 애니메이션 제작 팀이 접근가능하도록 모션 캡처 데이터를 단일 데이터 형태로 통합하여 저장함으로써 모션 캡처 데이터 사용을 쉽게 해주어야 하는 것이다. 이를 위해 XML과 ASP의 기술이 사용되었다. 그러나 본 논문의 연구와 다른 점은 이 논문에서 고려하는 모션 캡처 데이터들은 세그먼트 기반의 단순한 포맷으로 계층구조에 대해서는 고려하지 않고 있다. 또한 본 논문에서 제안한 MCML과 같은 모션 캡처 데이터를 위한 표준 마크업 언어도 제안하지 않고 있으며 XML을 이용한 데이터 변환에 대한 가능성만을 언급하고 있다.

둘째로, VHML(Virtual Human Markup Language)[8]은 W3C Voice Browser Activity에[9] 의해 표현된 것으로 XML/XSL 기반으로 이루어져 있다. VHML의 의미는 Pandzic(2001)과 같이 Talking Head/Talking Human (TH)을 사실적이고 더 자연스러운 상호작용이 되도록 해주는데 있다. 이 VHML은 모션 캡처 데이터와 직접적으로 관련 있는 것은 아니고 캐릭터의 감정 표현을 위한 얼굴 표정, 제스처, 신체를 이용한 감정 표현 등을 XML 기반의[10] 마크업 언어로 표현하는 방법을 제안하고 있다.

이 외에도 VRML(Virtual Reality Modeling Language)[11]은 웹에서 3차원 환경을 모델링하기[12] 위해 만들어진 언어이고 H-ANIM[13]은 Web3D 컨소시엄(Consortium)에서 확정된 표준 스펙으로 아바타를 3차원으로 모델링 하기 위한 구조를 기술하고 있다.

## 3. 모션 캡처 데이터 분석

모션 캡처 데이터 포맷은 모션 캡처 데이터의 데이터 처리 방법에 따라서 크게 트래커 포맷(Tracker Format)과 스켈레톤 포맷(Skeleton Format)으로 나눌 수 있다. 전자는 3차원 위치 값을 가지고 있는 형태로 Adaptive Optics AOA, National Institute of Health

C3D, Motion Analysis TRC 형태들이다. 후자는 스켈레톤 정보까지 가지고 있는 데이터 형태로서 Biovision hierarchical BVH, Biovision BVA, Biovision/Alias ASK/SDL, Motion Analysis HTR, Acclaim ASF/AMC, Acclaim-Motion Analysis SKL/MOT, Lamsoft Magnetic format BRD, Polhemus DAT files, Ascension ASC files 형태이다[14]. 이 파일 형태들은 다양한 데이터의 파일 구조를 가지고 있으므로 이 중에서 일반적으로 많이 사용되는 몇 개의 파일 구조를 살펴보고자 한다.

Biovision hierarchical BVH의 .bvh 파일 포맷은 원래 모션 캡처 데이터를 서비스하는 회사인 Biovision에 의해 개발되었고, 이 포맷은 모션 데이터뿐만 아니라 스켈레톤 계층 정보도 제공해 주고 있다[15]. Motion Analysis Corporation의 모션 캡처 데이터의 파일은 .trc 형태이다. Acclaim Motion Capture System의 파일 포맷 .asf는 조인트(joint) 회전 데이터를 기반으로 joints 와 bones의 형태, 계층구조, 특성들에 의해 스켈레톤으로 정의된다[16]. 이러한 데이터 파일 구조 형태를 살펴보면, 모션 데이터뿐만 아니라 스켈레톤 계층 정보도 제공해 주고 있다. 이 파일들은 프레임에 해당하는 모든 마커의 3차원 좌표 값을 가지고 있으며, 동작의 인체 계층 구조는 23 세그먼트 시스템으로 구성되어 있다.

BioVision사의 BVH 포맷은 HIERARCHY SECTION과 MOTION SECTION으로 나뉘어져 있다. HIERARCHY SECTION은 아바타의 골격 구조를 정의하는 부분이다. BVH에서 정의된 골격 구조는 총 18개의 조인트로 이루어져 있으며 Hips를 몸체의 Root로 하여 왼쪽 하체와 오른쪽 하체 그리고 상체방향으로 일련의 순서를 가지고 각각의 Segment를 JOINT로 연결한 방식으로 구성되어 있다. MOTION SECTION은 각 JOINT에 적용되는 오일러 각으로 구성되어 있다.

## 4. MCML(Motion Capture Markup Language)

현재 XML은 웹뿐만 아니라 기업 비즈니스 통합, 전자 문서, 교육 등 데이터를 처리하는 모든 분야에서 핵심 기술로 대두되고 있다. 이는 XML이 서로 다른 형식의 데이터들을 통합하고 교환하기 위한 표준화된 방법으로 인식되고 있으며 또한 이를 지원하는 상용 제품들이 계속적으로 개발되고 있기 때문이다. 그러므로 XML을 모션 캡처 데이터 관리에 응용할 경우 여러 애니메이터들이 서로 다른 포맷의 모션 데이터를 사용하여 공동으로 작업 할 때 높은 생산성과 효율성을 얻을 수 있다.

이를 위해서는 먼저 서로 다른 모션 데이터를 통합할

수 있는 하나의 표준화된 표현 방법이 있어야 한다. 본 논문에서는 XML(eXtensible Markup Language)를 이용하여 서로 다른 포맷의 모션 캡처 데이터를 하나의 공통된 폼으로 통합할 수 있는 MCML(Motion Capture Markup Language)를 제안한다. MCML은 모션 캡처 데이터의 헤더 정보, 관절체 정보, 모션 정보를 표현할 수 있는 태그 집합을 가지며 정확한 표현을 위한 DTD(Document Type Definition)을 가진다. 이번 장에서는 이들 데이터를 MCML이 어떤 방법으로 모션 캡처 데이터를 표현하고 통합하는지를 설명하고 있다.

#### 4.1 관절체 데이터

모션 캡처 데이터 파일에 들어 있는 데이터 중에 인

체의 관절을 표현하는 데이터가 있다. ASF/AMC, BVH, HTR 세 가지 포맷을 살펴보면 인체의 관절을 표현하는 방법과 관절에 대한 명칭이 각각 다르며 표현하는 정도도 다를 수 있다. 이를 하나로 통합하기 위해서는 먼저 각 데이터 포맷에서 인체 관절을 표현하는 방법 및 명칭을 조사하여 공통된 것끼리 매핑을 시키고 서로 간에 매핑되지 않는 부분은 추가시킨 다음 하나의 표준 명칭을 정의하는 것이다.

표 1은 이 세 파일 포맷의 매핑관계와 인체 관절을 표현하는 MCML의 태그들을 보여주고 있다.

ASF/AMC 파일 포맷에서 관절체를 표현하는 방법은 인체 뼈(bone)에 대한 명칭을 사용하는 것이고 BVH 파일

표 1 인체관절을 표현하기 위한 ASF/AMC, BVH, HTR 파일 포맷의 명칭과 MCML의 명칭간의 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 (lhumeral)	leftshoulder	lhumeral	lupperarm
7	left_low_arm	h_left_low_arm (lradius)	leftelbow	lradius	llowarm
8	left_wrist	(lwrist)	undefined	undefined	undefined
9	left_hand	h_left_hand (lhand)	leftwrist	lwrist	lhand
10	left_fingers	h_left_fingers (lfingers)	undefined	undefined	undefined
11	left_finger_one	h_left_finger_one (lthumb)	undefined	undefined	undefined
12	left_finger_two	h_left_finger_two	undefined	undefined	undefined
13	left_finger_three	h_left_finger_three	undefined	undefined	undefined
14	left_finger_four	h_left_finger_four	undefined	undefined	undefined
15	left_finger_five	h_left_finger_five	undefined	undefined	undefined
16	right_shoulder	h_right_shoulder (rclavicle)	rightcollar	rshoulderjoint	undefined
17	right_up_arm	h_right_up_arm (rhumeral)	rightshoulder	rhumeral	rupperarm
18	right_low_arm	h_right_low_arm (rradius)	rightelbow	rradius	rlowarm
19	right_wrist	(rwrist)	undefined	undefined	undefined

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

포맷에서는 마커의 위치에 대한 명칭을 사용한다. 그리고 HTR은 인체 뼈에 대한 명칭을 사용하지만 초창기에 나온 포맷이므로 적은 수의 명칭만을 가진다. 이들 파일을 통합하기 위해 MCML에서는 ASF/AMC를 기반으로 이 세 포맷을 전부 포함할 수 있도록 확장된 표현력을 가진다. 위의 표 1에 있는 내용은 본 논문에서 구현한 시스템에서 서로 다른 파일과 MCML간의 매핑을 위한 테이블로 사용된다. 즉 BVH 파일 하나가 입력으로 들어오면 파일을 파싱하면서 관절체의 이름을 이 매핑 테이블에서 탐색하여 대응하는 MCML 명칭을 찾아 온 다음 이를 이용하여 BVH 파일에 대응하는

MCML 파일을 생성한다.

#### 4.2 헤더 데이터

모션 데이터 파일에는 파일 타입, 버전, 주석 등 부가적인 정보를 담고 있는 헤더 데이터 영역이 있다. 물론 BVH와 같이 헤더 정보가 들어 있지 않은 파일도 있다. MCML에는 이들 파일의 헤더 정보를 모두 포함할 수 있는 태그 집합을 제공한다. 표 2에 헤더 데이터의 매핑 관계가 나와있다.

헤더 정보중에는 모델링되는 캐릭터 구조나 모션과 직접적인 연관이 없는 것들도 많이 있으며 향후 버전에서 사용될 수 있도록 미리 명칭만 정의해 놓은 것도 있다.

표 2 모션 캡처 데이터의 헤더 정보를 나타내기 위한 BVH, ASF/AMC, HTR 파일의 명칭과 MCML의 명칭 간의 매핑관계

	MCML	ASF/AMC	BVH(1)	BVH(2)	HTR
1	filetype	undefined	undefined	undefined	FileType
2	datatype	undefined	undefined	undefined	DataType
3	filename	undefined	undefined	undefined	undefined
4	version	version	undefined	undefined	FileVersion
5	skeleton_name	name	undefined	undefined	undefined
6	units Attribute: mass length angle	units Attribute: mass length angle	undefined	undefined	undefined
7	num_segments	undefined	undefined	undefined	NumSegments
8	num_frames	undefined	undefined	undefined	NumFrames
9	dataframe_rate	undefined	undefined	undefined	DataFrameRate
10	euler_rotation_order	undefined	undefined	undefined	EulerRotationOrder
11	calibration_unit	undefined	undefined	undefined	CalibrationUnits
12	global_axis_of_gravity	undefined	undefined	undefined	GlobalAxisogGravity
13	bone_length_axis	undefined	undefined	undefined	BoneLengthAxis
14	scale_factor	undefined	undefined	undefined	ScaleFactor

MCML에서는 이 모든 정보를 유지할 수 있는 헤더 정보를 위한 태그 집합을 그림 1처럼 정의하고 있으며 새로운 헤더 정보가 추가로 정의될 수도 있다.

```

<!--Header Information -->
<header>
  <filetype> htr </filetype>
  <datatype> HTRS </datatype>
  <filename> quickwalk.htr </filename>
  <version> 1.0 </version>
  <skeleton_name> Billy </skeleton_name>
  <units mass="1.0" length="0.45" angle="degree"/>
  <num_frames> 60 </num_frames>
</header>
    
```

그림 1 MCML로 표현된 헤더 정보

4.3 관절체 계층 데이터

모션 캡처 데이터에는 모델링되는 캐릭터에 대한 정보가 들어 있다. 즉 모션을 적용할 캐릭터가 같은 파일

안에 정의되어 있는 것이다. 물론 Acclaim의 ASF/AMC 모션 캡처 데이터 파일은 캐릭터와 모션을 별도로 관리하도록 한다. ASF 파일에 캐릭터의 관절체 및 계층, 그리고 초기 위치 정보가 들어있고 AMC 파일에는 이 캐릭터가 움직이는 모션 정보가 들어 있다. 이처럼 캐릭터와 모션을 분리함으로써 얻을 수 있는 장점은 동일한 모션을 유사한 사이즈와 스�কে틀론을 가지는 캐릭터에 같이 적용할 수 있다는 것과 캐릭터가 재사용 가능하다는 것이다. 그러나 BVH, HTR과 같은 파일 포맷은 캐릭터 정보와 모션 정보가 하나의 파일안에 있어 야만 하므로 재사용성이 떨어진다. MCML에서는 인체 관절 정보, 헤더 정보, 관절 계층정보, 모션 정보등을 분리할 수 있고 또 각각의 일부분들을 조합하여 하나의 파일을 만들 수도 있으므로 재사용성이 높으며 중복 작업이 없으므로 애니메이터에게 많은 도움을 줄 수 있다. 표 3에서는 이들 파일들과 MCML간의 관절 계층 정보에서의 매핑 관계를 보여주고 있다.

관절체 계층 정보는 캐릭터를 모델링 하기 위한 실제적인 정보로서 루트에서부터 인체를 모델링 하기 위한

표 3 모션 캡처 데이터의 관절체 계층 정보를 나타내기 위한 ASF/AMC, BVH, HTR 파일의 명칭과 MCML의 명칭간의 매핑관계

	MCML	ASF/AMC	BVH	HTR
1	root Attribute: order axis position orientation	root Attribute: order axis position orientation	root Attribute: channels channels offset	undefined
2	Skeleton	bonedata	hierarchy	
3	bone Attribute: id	bonedata Attribute: id	hierarchy	SegmentName & Hierarchy
4	Name	name	undefined	
5	direction	direction	undefined	
6	Length	length	undefined	
7	Position	position	undefined	
8	Axis	axis	offset	
9	Order	order	channels	
10	Dof	dof	undefined	
11	Limits	limits	undefined	
12	bodymass	bodymass	undefined	
13	Cofmass	cofmass	undefined	

```

<!-- Skeleton Information -->
<root order="Tx Ty Tz Rx Ry Rz" axis="XYZ" position="0 0 0"
orientation="0 0 0"/>
<skeleton>
  <bone id="1">
    <name> lfemur </name>
    <direction> 0.34202 0.939633 0 </direction>
    <length> 6.8845 </length>
    <axis> 0 0 20 XYZ </axis>
    <dof> rx ry rz </dof>
    <limits> -160.0 20.0 </limits>
    <limits> -70.0 70.0 </limits>
    <limits> 60.0 70.0 </limits>
  <bone id="2">
    <name> ltibia </name>
    ...
  <bone id="3">
    <name> lfloor </name>
    ...
  </bone>
</bone>
</bone>
</skeleton>

```

그림 2 관절체 구조의 예를 보이는 MCML 문서의 일부분. 인체의 각 뼈는 bone 엘리먼트로 표현 되고 뼈들 간의 계층구조는 bone 엘리먼트들간의 관절 계층구조로 표현된다.

각 관절 뼈 부분의 상대적인 위치와 관절각 정보를 가지고 있다. 이 계층정보에 오류가 있을 경우 모델링되는 캐릭터가 이상한 모습으로 보여진다.

MCML에는 인체 캐릭터를 모델링 하기 위해 root와 각 관절의 계층 정보 및 관절 길이, 상대적인 거리 등의 정보를 나타낼 수 있는 엘리먼트와 애트리뷰트 집합이 들어 있다. 표 3에 나와있는 MCML의 skeleton 엘리먼트의 name 애트리뷰트는 각 관절의 이름을 가지는 것으로 이 이름은 표 1에서 정의한 인체 구조 명칭을 사용해야 한다.

그림 2의 MCML문서는 캐릭터 관절 계층 구조의 한 예를 보여주고 있다. root는 캐릭터의 초기 위치를 가리키고 skeleton은 각 관절의 계층구조와 위치, 크기 등을 가리킨다. 계층구조는 bone 엘리먼트의 내포관계를 이용하여 표현할 수 있다.

**4.4 모션 데이터**

인체 캐릭터가 모델링 되면 모션 캡처 데이터 파일의 모션 데이터를 이용하여 동작을 일으킨다. 모션 데이터는 전체 프레임 수, 한 프레임 당 시간, 그리고 각 프레

임 마다의 이동(translation), 회전(rotation) 값들을 가진다.

모션 캡처 데이터 포맷에서의 모션 데이터는 공백문자로 분리되어 연속적으로 나열되어 있기 때문에 모션 데이터의 일부 또는 몇 개의 프레임만을 따로 분리해 내기가 어렵다. 모션 편집에서도 특정 프레임을 분리해 내거나 다른 프레임과 위치 변경 또는 일부 관절의 움직임만을 재조정하기가 어렵다.

MCML에서는 각 프레임을 별도로 분리하고 또한 한 프레임의 관절들도 별도로 분리함으로써 특정 프레임 또는 특정 관절 모션들을 추출하기가 쉽고 위치 변경 또는 결합, 분리가 가능하다. MCML의 frame 엘리먼트의 하위 엘리먼트인 frame\_bone에는 그 프레임에서의 각 관절의 translation과 rotation 값을 가진다. motion\_name에는 이 한 프레임에서 각 관절의 움직임을 통해 보여지는 모션에 대한 이름을 부여한다. 모션 캡처 데이터가 저장장치에 저장되어 있다고 해도 특정 모션을 검색하기는 쉽지 않다. 특히 MCML 구조가 아닌 모션 캡처 데이터 포맷 그대로 저장할 경우에는 뷰어로 그 데

이타를 실행해 보기 전에는 모션 데이터만으로 어떤 모션을 취하는지 알기가 어렵다.

모션 데이터를 MCML로 변환하여 저장할 경우 XPATH나 XQL과 같은 XML 질의어를 통하여 프레임 별로 검색이 가능하다. 이때 원하는 모션을 검색할 수 있도록 각 프레임마다 해당되는 그 모션을 알려주는 모션명을 같이 저장함으로써 이 모션명을 이용한 검색이 가능한 것이다.

애니메이터는 새로운 캐릭터를 만든 다음 이 캐릭터에게 모션을 적용하기 위해 모션 캡처를 새로 받을 필요없이 저장된 모션 데이터 중에서 원하는 모션을 모션명(예. 왼팔을 벌리는 모션일 경우 left-arm-abduction 이라 한다)을 이용하여 검색한 다음 이 모션 데이터를 변경함으로써 새로운 캐릭터에게 쉽게 적용할 수 있다.

표 4 모션 캡처 데이터의 모션 정보를 나타내기 위한 ASF/AMC, BVH, HTR 파일의 명칭과 MCML의 명칭간의 매핑관계

	MCML	ASF/AMC	BVH	HTR
1	Motion		motion	
2	Frames		frames	
3	frametime		frametime	
4	frame Attribute: id	frame Attribute: - Tx Ty Tz Rx Ry Rz	frame Attribute: #Fr Tx Ty Tz Rx Ry Rz	frame Attribute: frame# Tx Ty Tz Rx Ry Rz
5	frame_bone Attribute: name Tx Ty Tz Rx Ry Rz			
6	motion_name			

모션을 표현하기 위한 인체의 움직임을 나타내는 표준 용어로 다음의 것들이 있다.

flexion : 구부리는 동작, 손가락을 구부리다. 허리를 구부리다

extension : 펴는 동작, 손가락을 펴다. 팔을 펴다. 허리를 펴다

abduction : 벌리다. 팔을 벌리다. 다리를 벌리다

adduction : 오므리다. 팔을 오므리다. 다리를 오므리다

medial(internal) rotation : 안으로 돌리다

lateral(external) rotation : 바깥으로 돌리다

left or right rotation : 목이나 몸을 좌우로 돌리다

이 외에도 다리, 몸통, 머리, 어깨 등 신체 각 부위의 움직임을 정의하고 있다. 이때 중요한 것은 불가능한 동작을 표현하는 것이 없어야 한다는 것이다. 예를 들어 left-shoulder-abduction(어깨를 벌리다)라는 동작은 일어날 수 없는 동작이므로 이러한 모션명은 정의하지 않아야 한다.

```

<motion>
  <frames> 89 </frames>
  <frametime> 0.3333 </frametime>
  <frame id="1">
    <motion_name> left-arm-front </motion_name>
    <frame_bone name="hips"
      Tx="-5.61" Ty="37.37" Tz="-68.55"
      Rx="-1.9" Ry="4.16" Rz="-5.28"/>
    <frame_bone name="chest"
      Tx="-5.61" Ty="37.37" Tz="-68.55"
      Rx="-1.9" Ry="4.16" Rz="-5.28"/>
    ...
  </frame>
  ...
</motion>
  
```

그림 3 캐릭터 모션의 예를 보이는 MCML 문서의 일부분. frame 엘리먼트에 한 프레임 정보로서 프레임 시간, 모션 이름이 들어 있으며 frame\_bone 엘리먼트는 해당 프레임에서의 각 관절의 위치 및 회전에 대한 값을 가진다.

### 5. MCML 기반의 모션 데이터 관리

#### 5.1 MCML 기반의 시스템 프레임워크

이 장에서는 MCML 기반의 모션 캡처 데이터 관리 시스템의 프레임워크에 대해 설명한다. 시스템은 크게 MCML을 관리하고 저장하는 시스템과 MCML을 이용한 모션 캡처 데이터 기반의 모션 편집 및 리타겟팅 시



시스템으로 구성된다. MCML을 관리하고 저장하는 시스템은 ASF/AMC, BVH, HTR 등의 여러 모션 캡처 데이터 포맷 파일들을 MCML 포맷 문서로 변환하거나 MCML을 다시 모션 캡처 데이터 포맷으로 역변환하는 부분과 MCML 문서의 저장과 검색을 포함한다. 그리고 XML 편집기를 이용하여 MCML 문서 자체를 편집할 수 있으며 이렇게 편집된 내용을 다시 ASF/AMC, BVH, HTR 등으로 변환한 다음 편집한 내용을 애니메이션 소프트웨어를 이용하여 모션을 확인해 볼 수 있다.

모션 편집 및 리타겟팅 시스템은 이미 캡처된 모션 데이터를 새로 만든 또는 기존의 캡처된 다른 캐릭터에 적용하여 새로운 모션을 만들거나 모션 리타겟팅을 하도록 지원하는 시스템이다. 이러한 캡처 데이터 기반의 모션 편집과 리타겟팅을 위해서 가장 중요한 것은 원하는 모션을 어떤 방법으로 찾아내는가 하는 것이다. 그리고 검색된 일부 모션들을 다른 스케일과 다른 조인트를 가지는 캐릭터에 적용할 경우 자연스런 모션을 생성할 것인가 하는 것이다. 이 부분은 아직 구현되지 않은 부분으로 현재 연구가 진행중이다. 그러므로 본 논문에서는 이 부분에 대해 간단히 소개하고 넘어가도록 하겠다.

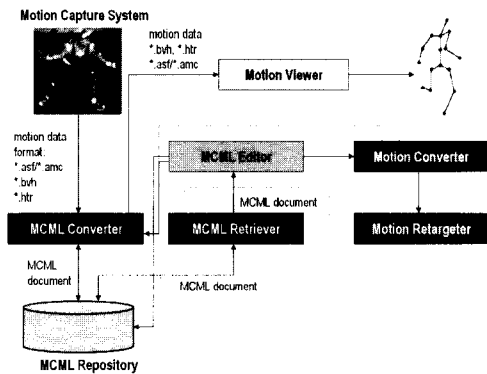


그림 4 MCML을 기반으로 모션 캡처 데이터를 변환, 검색, 편집, 재가공 및 리타겟팅을 하기 위한 시스템 구성도

MCML Repository는 모션 캡처 데이터를 변환한 MCML 문서를 저장하는 장치이다. MCML이 XML 문서이기 때문에 XML 전용 데이터베이스를 사용하거나 Oracle, MS SQL Server와 같은 상용 데이터베이스를 사용할 수 있다. 간단히 구현하기 위해서는 파일 시스템

에 저장할 수도 있다. 본 논문에서는 GMD-IPSI XQL Engine[17]을 이용하여 MCML 문서를 저장한다. GMD-IPSI XQL Engine 은 XML 문서의 저장뿐만 아니라 문서의 검색을 지원한다.

5.2 MCML 문서 변환

모션 캡처 데이터 파일과 MCML 사이의 변환을 담당하는 MCML Converter 모듈은 Motion Capture Data Parser와 MCML Wrapper로 구성된다. 모션 캡처 데이터 파일이 입력으로 들어오면 먼저 Motion Capture Data Parser에서 이 파일의 토큰들을 분석해 낸 다음 이들을 MCML Wrapper로 전달한다. MCML Wrapper는 이 토큰들을 이용하여 MCML 문서를 생성한다. 이때 bone naming mapping 테이블이 사용된다. 이 매핑 테이블에는 표 1에서 보여진 MCML과 각 모션 캡처 데이터간의 관절 명칭에 대한 매핑정보를 가지고 있는 테이블이다.

Motion Wrapper에서는 토큰으로 들어 온 모션 캡처 데이터의 관절명을 이에 매핑되는 MCML 관절명을 찾은 다음 이 명칭을 관절 계층 구조를 작성할 때 name 애트리뷰트의 값으로 지정한다. Motion Wrapper는 ASF/AMC\_to\_MCML, BVH\_to\_MCML, HTR\_to\_MCML로 구성되어 있다. 생성되는 MCML 문서의 구조와 엘리먼트 및 애트리뷰트들은 표 2, 3, 4의 정의를 따른다.

MCML 문서를 생성하기 위해 mcml.dtd 라는 DTD(Document Type Definition) 문서를 사용한다. DTD는 정당한 XML 문서를 생성하기 위한 문법으로서 XML 문서의 구조에 대해 정의하는 것이다. mcml.dtd 에는 MCML 문서가 가져야하는 엘리먼트들간의 계층 관계와 애트리뷰트 포함 및 필수 항목과 선택 항목에 대해 정의하고 있다.

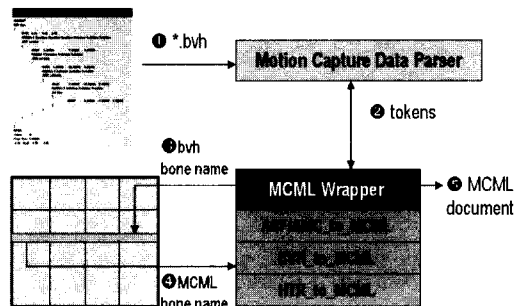


그림 5 BVH 포맷의 모션 캡처 데이터 파일을 대응하는 MCML 문서로 변환하기 위한 프로세스

새로운 모션 캡처 데이터 포맷에 대한 MCML 변환이 요구되는 경우에는 매핑 테이블에 없는 관절 명칭을 매핑 테이블에 추가한 다음 MCML Wrapper에 변환 함수를 추가하기만 하면 된다.

### 5.3 MCML 문서의 편집과 모션 뷰어

생성된 MCML 문서는 리퍼지토리(Repository)에 저장된다. 리퍼지토리에는 저장된 XML 문서 리스트와 각 문서의 내용을 조회할 수 있는 기능을 가진다. 변경이 필요한 문서는 먼저 그 문서를 리퍼지토리에서 검색한 다음 MCML 편집기에서 편집이 가능하다. 편집기에는 MCML 문서를 읽어 들인 다음 트리 구조와 그래프 구조를 보여 주고 텍스트 형태로 내용을 보여 준다.

모션 캡처 데이터 파일에서 뿐만 아니라 이 MCML 문서에서 새로운 프레임을 생성하거나 캐릭터의 형태를 변경할 수 있다. 모션 캡처 데이터 파일은 계층 구조를 보이지 않는 텍스트 형태의 문서이므로 내용을 분석하거나 변경하기가 쉽지 않다. 그러나 MCML 문서는 계층구조를 가지고 있으며 중요 정보들이 분리되어 있기 때문에 쉽게 변경이 가능하다.

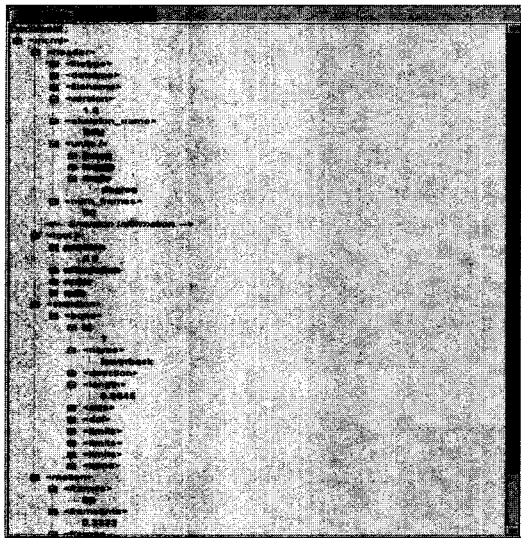


그림 6. MCML 샘플 문서의 트리 구조

그림 6에서는 MCML 한 문서를 편집기에서 트리 구조로 보여주는 것을 나타낸다. 이 트리구조를 보면 입력으로 들어 온 MCML 문서에 캐릭터의 관절 구조가 어떤 형태로 되어 있으며 각 프레임의 동작과 모션 이름은 어떻게 되는지 쉽게 확인 할 수 있고 또한 편집도 쉽게 할 수 있다. 편집기를 통하여 변경된 MCML 문서

는 MCML로 리퍼지토리에 다시 저장할 수 있고 모션 캡처 데이터 포맷으로 export 시킬 수도 있다. 모션 캡처 데이터 포맷으로 변경된 파일은 애니메이터가 가지고 있는 애니메이션 소프트웨어에서 모션을 실행할 수 있고 재 가공도 가능하다.

본 논문의 시스템에서는 모션 캡처 데이터 파일을 입력으로 받아서 실제 캐릭터의 모션을 보여주는 모션 뷰어를 제공한다.



그림 7 모션 캡처 데이터를 입력받아 모션을 보여 주는 뷰어

본 논문의 시스템은 MCML 문서를 직접 분석하여 캐릭터 모션을 보여주지는 않는다. 이것은 여러 애니메이터들이 모션 데이터를 공유하면서 같이 작업을 할 경우 각 애니메이터 마다 서로 다른 애니메이션 소프트웨어를 사용할 수 있다. 이때 각 소프트웨어에 MCML을 이해할 수 있는 플러그 인을 따로 개발할 필요 없이 각 소프트웨어에서 이해할 수 있는 모션 캡처 데이터 포맷으로 익스포트시켜서 그 파일을 각 사용자가 다운받아 자신의 시스템에서 확인하는 것이 더 유용하기 때문이다.

## 6. 결론 및 향후 연구 계획

3차원 애니메이션에서 동작을 생성하는 기법으로 모션 캡처 방식은 실제 인체 모션에 가까운 좋은 품질의 캐릭터 모션을 얻을 수 있다는 점에서 애니메이션 제작을 위해 널리 사용되고 있다. 그러나 사용이 복잡하고 비용이 비싼 장비로 인해 필요한 모든 모션을 모션 캡처에 의해 생성하는 것은 비용측면에서 볼 때 비생산적이고 경우에 따라서는 불가능한 경우도 있다. 그러므로 애

니메이터 입장에서는 과거에 캡처한 모션 데이터를 재활용함으로써 약간 다른 모션을 만들거나 여러 모션을 합성해서 새로운 모션을 만들어 낼 수 있기를 원한다.

본 논문은 이러한 모션 캡처 데이터의 재사용성에 대해 초점을 맞추고 있다. 요즘은 영화나 게임 산업에서 모션 캡처 데이터가 현실적인 동작을 생성하기 위한 필수적인 방법으로 등장하면서 여러 애니메이터들이 모션 데이터를 공유하거나 서로 교환하는 일이 요구된다. 그러나 모션 캡처 데이터는 어떤 모션 캡처 시스템을 사용하느냐에 따라서 데이터 포맷에 많은 차이를 보인다. 이러한 차이는 애니메이터가 모션 캡처 데이터를 가지고 작업하는데 어려움을 주는 한 가지 주요한 원인이다. 즉 애니메이터는 자신에게 익숙한 모션 편집 소프트웨어가 지원하지 않는 포맷의 모션 데이터를 받았을 경우 이 포맷을 인식할 수 있는 플러그인을 따로 구하거나 아니면 현재 자신의 소프트웨어에서 처리 가능한 포맷으로 변환하여야 한다.

이에 대한 해결책으로 본 논문에서는 현재 데이터 통합 및 교환을 위한 포맷으로 널리 활용되고 있는 XML(eXtensible Markup Language)을 이용해서 모션 데이터에 대한 표준 포맷을 제공하는 방법에 대해 제안하고 있다. 그리고 이 표준 포맷을 이용하여 모션 검색 및 편집이 가능한 시스템이 가져야 하는 주요 구성요소와 기본 구조를 보여주는 프레임워크를 제안하였다.

여러 가지 포맷의 모션 캡처 데이터의 주요 스펙을 분석하여 공통적인 요소들과 각 포맷마다의 특이사항들을 모아서 표준적인 태그집합을 정의하였으며 캐릭터 정보와 모션 정보를 나타내기 위한 XML 문서 구조를 정의하였다. 이렇게 정의된 XML 문서 구조를 MCML(Motion Capture Markup Language)라고 한다.

서로 다른 포맷의 모션 캡처 데이터를 먼저 MCML 문서 구조에 따라 MCML 문서로 변환한 다음 이 문서를 XML 데이터베이스에 저장한다. 저장된 MCML 문서는 언제든지 검색이 가능하고 엘리먼트 수준에서 검색이 가능하므로 특정 모션을 쉽게 검색할 수 있다. 애니메이터는 자신이 원하는 모션을 검색한 다음 다른 캐릭터에게 그 모션을 적용할 수도 있고 여러 모션들을 합성하여 연결된 동작을 행하도록 만들 수도 있다.

본 연구는 크게 모션 데이터의 변환 및 통합 부분과 저장된 모션 클립들을 조합하여 새로운 모션을 생성하는 모션 캡처 데이터에 기반한 모션 편집으로 나뉘어진다. 본 논문에서는 모션 데이터의 변환과 통합을 위한 시스템을 소개하였다. 향후에는 저장된 모션 클립들을 효율적으로 검색하는 방법과 검색된 이 클립들을 조합

하여 새로운 모션을 생성하거나 다른 캐릭터에게 저장된 모션을 적용함으로써 동일한 모션을 행하도록 하는 모션 리타겟팅에 대해 연구하고자 한다.

이러한 연구가 완성될 경우 3차원 애니메이션 제작 분야에서 모션 캡처 데이터를 이용할 경우 데이터의 재사용성이 높아짐으로써 비용 절감 및 보다 좋은 품질의 애니메이션을 제작할 수 있을 것으로 기대한다.

## 참 고 문 헌

- [1] Zoran Popovic and Andrew Witkin, "Physically Based Motion Transformation", Proceedings of SIGGRAPH 99, Los Angeles, August 8-13, pp. 11-20, 1999.
- [2] D. Tolani and N. I. Badler, "Real-time inverse kinematics of the human arm", Presence, 5(4), pp. 393-401, 1996.
- [3] 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, pp. 3-18, 1997.
- [4] Chronopolus, Emilio. Personal Interview. Character Animator. NuFX, Rolling Meadows: Illinois, 1999.
- [5] Fiore, Richard. Personal Interview. Art Director. High Voltage Software, Hoffman Estates: Illinois, 1999.
- [6] Newtek. LightWave 3D 5.5 Users Manual. Austin, Texas: Newtek, 1998.
- [7] Morales, C.R., Development of an XML Web Based Motion Capture Data Warehousing and Translation System for Collaborative Animation Projects (USA), WSCG 2001.
- [8] VHML, "Virtual Human Markup Language. Online at <http://vhml.org/>", 2001
- [9] Mahoney, Diana. Face Tracking. Computer Graphics World, Vol. 20, Issue 4, pp 23-30, Apr, 1997.
- [10] W3C, "Extensible Markup Language (XML) 1.0. Online at <http://www.w3.org/XML/>", 1997.
- [11] Web 3D Consortium, "VRML 97 Specifications International Standard. Online at <http://web3d.org/technicalinfo/specifications/vrml 97>", 1997
- [12] Discreet. 3D Studio 3.0 Users Manual. San Rafael, CA, 1999.
- [13] Bernie Roehl, "Specification for a Standard Humanoid Version 1.1. Online at <http://www.h-anim.org/>", 1999
- [14] Alias Wavefront. Maya 2.0 Users Manual. Toronto, Canada, 1999.
- [15] Biovision Incorporated, "Biovision Motion Data Specifications. Online at <http://www.biovision.com/>",

1999.

- [16] Acclaim Incorporated, "Acclaim ASF/AMC File Specifications. Online at <http://www/acclaim.com>", 1999.
- [17] GMDIPS, "GMD-IPSI XQL Engine Version 1.0.2, Online at <http://xml.darmstadt.gmd.de/xql/>", 1999.



정 현 숙

1993년 대구가톨릭대학교 물리학과 학사 (B.S.). 1995년 대구가톨릭대학교 전산학과 석사(M.S.). 2000년 ~ 현재 연세대학교 컴퓨터학과 박사과정. 2002년 ~ 현재 안양대학교 디지털미디어학부 강의전담교수. 관심분야는 멀티미디어 응용, 멀티미디어 정보검색, 게임제작 기술 등



이 일 병

1976년 연세대학교 전자공학과 공학사. 1983년 University of Illinois, M.S. (Computer Science). 1986년 University of Massachusetts, Ph.D. (Computer and Information Science). 연세대학교 전산학과 과장 역임. 연세대학교 소프트웨어응용연구소 소장 역임. 한국정보과학회 인공지능연구회 회장 역임. 한국정보과학회 신경회로망연구회 회장 역임. 한국정보과학회 이사 역임. 한국인지과학회 이사 역임. 한국 퍼지및지능시스템학회 이사 역임. 2001년 한국인지과학회 회장. 2001년 현재 한국데이터마이닝학회 부회장. 1985년 이후 ~ 현재 연세대학교 컴퓨터학과 교수. 관심분야는 인공지능, 문자인식, 데이터마이닝, 컴퓨터비전, 홍채인식 등