

XML 기반의 FSM 시스템에 관한 연구

이정훈*^o 신성운** 오상권*

*호서대학교 컴퓨터공학부 **호서대학교 벤처대학원

zidkenu@hotmail.com melcaun@netian.com ohsk@office.hoseo.ac.kr

A Study of XML-based FSM Definition System

JungHoon Lee*^o SungWoon Shin** SamKweon Oh*

* School of Computer Engineering, Hoseo University

** Dept. of The Graduate School of Venture, Hoseo University

요 약

가상공간에는 PC(Playerable Character), NPC(Non-Playerable Character)등의 동적 객체와 건물, 지형 등의 정적 객체들이 존재하게 된다. 동적 객체들의 경우, 현실감을 위해 인공지능이 자주 이용된다. 현재 까지 인공지능에 대한 연구는 유한상태기계(Finite State Machine, FSM), 학습 알고리즘, 유전자 알고리즘, 신경망 알고리즘 등을 중심으로 진행되어 왔다. 이중 유한상태기계는 비교적 알고리즘이 간단하고, 시스템의 부담이 적어 간단한 객체의 인공지능으로 가장 널리 사용되고 있다. 본 논문은 유한상태기계를 확장하여 모드변경(Mode Change)과 그룹행동을 보여줄 수 있는 XML을 활용한 FSM 시스템을 제안한다. 여기서 모드변경이란 하나의 행동 패턴에서 다른 행동 패턴으로 변경하는 것을, 그룹행동은 여러 객체가 함께 행동하는 Flocking기법을 지칭한다. 이러한 XML을 활용한 FSM 시스템은 다양한 패턴의 정의는 물론, 객체의 상태 정의 및 수정, 확장이 용이하여, 다양한 응용 분야에서 활용될 수 있다.

1. 서 론

인터넷 기술이 발전함에 따라 여러 사용자가 공간을 공유할 수 있는 가상공간에 대한 관심이 높아졌다. 가상공간은 현실세계를 컴퓨터를 이용하여 가상의 공간에 리모달링한 공간이며, 지리적으로 떨어져 있는 사용자들이 네트워크를 이용하여 임의의 행동이나 공동 작업이 이루어지는 공간이다.[1,2]

최근에는 3차원 그래픽 기술의 발전과 함께 그래픽을 이용하여 현실감 있는 가상 환경을 만들고 있다. 이러한 가상공간에는 PC, NPC등의 동적 객체와 건물, 지형 등의 정적 객체들이 존재하게 된다. 동적 객체들의 경우, 현실감을 위해 인공지능이 자주 이용된다. 현재까지 인공지능에 대한 연구는 상태들을 정의하고 입력에 따라 상태를 변화시키는 유한상태기계(Finite State Machine, FSM), 객체가 스스로 학습하여 진화하는 학습 알고리즘 및 유전자 알고리즘이나 신경망 알고리즘을 이용한 인공지능, 객체들의 집단행동을 표현하기 위해 Flocking기법, A* 알고리즘을 이용한 길찾기 알고리즘 등을 중심으로 연구되어 왔다.[3]

그러나 Flocking기법은 여러 객체들의 그룹행동만을 표현하고, 길찾기 알고리즘은 가상공간에서 정적 객체들과의 충돌을 피해 길을 찾아가는데 이용된다. 또한, 신경망과 유전자 알고리즘을 이용한 인공지능과 같은 알고리즘은 구현이 어렵고 복잡하다. 그러나, FSM은 여러 상태들을 정의하고, 입력에 따른 출력을 통해 상태를 천이하므로, 알고리즘이 간단하고 빠르게 응답하며 개발이 용이한 장점이 있다. 또한, 서버에 부담이 적어 네트워크를 이용한 서버 클라이언트 구조에서 적합하다.

본 논문에서는 XML기반의 FSM 시스템을 설계한다. 본 시스템은 FSM과 XML을 활용하여 여러 행동 패턴들을 정의하여 객체의 행동 패턴이 변경되는 모드변경(mode change)의 표현이 가능하다. 더 나아가 여러 객체가 함께 행동하는 그룹행동의 표현도 가능하다. XML의 활용으로 객체의 상태 정의 및 수정이 용이하여, 확장성과 효율성이 뛰어나다. 본 논문의 구성은 다음과 같다. 2장에서 FSM을 이용한 관련 연구를 설명한다. 3장에서 전체 시스템 구조 및 FSM모듈과 XML 모듈을 설명하고

마지막으로 4장에서 결론을 맺는다.

2. 관련연구

가상공간에서 동적 객체들의 인공지능에 대한 연구는 유한상태기계(Finite State Machine, FSM), 학습 알고리즘, 유전자 알고리즘, 신경망 알고리즘 등을 중심으로 행해지고 있다. 이중 유한상태기계는 알고리즘이 간단하고 빠르게 응답하며 개발이 용이한 장점이 있다. 또한 서버에 부담이 적어 네트워크를 이용한 서버 클라이언트 구조에서 인공지능 구현을 위한 알고리즘으로 적합하다. 또한, 가상공간에서 객체들의 행동은 일정한 패턴들을 가지고 있어 인공지능, 신경망과 같은 알고리즘보다는 빠르게 응답하여 서버에 부담을 주지 않는 알고리즘이 효율적이다. FSM을 이용하여 모드 변경과 그룹 행동이 가능하므로 신경망이나 인공지능과 같은 알고리즘을 사용하지 않더라도 보다 지능적인 인공지능을 나타낼 수 있다. 현재까지 연구되어진 FSM을 이용한 인공지능 기법에는 Hierarchical FSM과 FuSM(Fuzzy State Machine)등이 있다.

2.1 Hierarchical FSM

Hierarchical FSM은 '예측(lookahead)'과 가정(what-if reasoning)을 기반으로 하여 상태들을 보다 체계적으로 표현한다. 상태들의 변화를 트리 구조로 나타냄으로 단순히 복잡해 질 수 있는 상태들을 보다 체계적이며 효율적으로 관리할 수 있고, 현 상태에서 천이될 수 있는 다음 상태들을 쉽게 파악할 수 있다.

그림 1은 간단한 Hierarchical FSM의 구조를 보여주고 있다. 그림 1에서 원은 각각의 상태들을 나타내며 우측의 레벨들은 각 상태에서의 레벨을 나타낸다. k레벨의 입력은 k+1레벨의 출력값과 정보(information)를 기반으로 한다. 또한, k레벨의 상태들은 k-1레벨의 상태들에 대한 간략한 정보들을 가지고 있다. 따라서 k레벨에서 k-1레벨로 천이하기 위한 예측과 가정이 가능하다. 또한 임의의 레벨에서의 상태 출력값이 '0'이라면 이

는 가상공간 전체에 어떤 변화에 대한(Update) 출력으로, 모든 객체에 전달된다.[4]

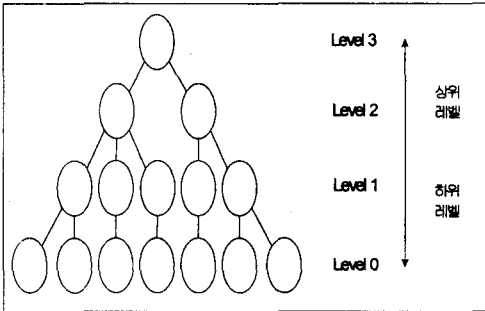


그림 1 Hierarchical FSM 구조

이러한 Hierarchical FSM의 장점은 상태 전이가 현 상태에서 전이될 수 있는 모든 상태를 검색하지 아니하고, 일정 범위의 상태만을 검색하여 빠르게 응답하며, 보다 적절한 다음 상태를 예측하여 전이 할 수 있다.

그러나 이러한 Hierarchical FSM은 현 레벨에서의 출력으로 상위레벨의 다른 상태로의 전이는 불가능하며, 트리구조의 다른 방향으로 전이가 불가능하여 패턴의 변화가 단순하다.

2.2 FuSM(Fuzzy State Machine)

FuSM은 상태기계(State Machine)와 Fuzzy이론이 더하여진 것이다. Fuzzy이론은 확률을 포함하는 비결정적인 것, 정확한 판단이 아닌 애매모호한 입력에 대하여 그에 따른 출력이 가능하다.

표 1 거리에 따른 입력 범위 표 2 감정에 따른 입력 범위

거리	값	감정	값
바로 여기	0.00 ~ 0.15	없음	0.00 ~ 0.20
매우 가까움	0.10 ~ 0.35	약간 싫어함	0.10 ~ 0.50
가까움	0.30 ~ 0.70	싫어함	0.40 ~ 0.80
조금 멀	0.60 ~ 0.90	매우 싫어함	0.75 ~ 1.00
매우 멀	0.85 ~ 1.00		

표 1은 거리에 따른 범위를 나타내고 있으며, 표 2는 다른 객체에 대한 감정의 범위를 나타낸다.

표 1과 표 2를 근거로 하여 객체는 거리가 얼마만큼 떨어져 있고 감정이 어느 범위에 있을 때 어떤 행동을 해야 하는지를 판단하게 된다.[5]

이러한 FuSM은 참과 거짓(0과 1)으로만 표현하는 것이 아닌 연속적인 값(0.0 ~ 1.0)으로 세분화한다. 이 범위 값에 기반하여 행동을 표현함으로써 참과 거짓이냐의 이분법 적인 출력이 아닌 유동적인 출력이 가능하다. 거리의 값이 0.13이라면 '바로 여기' 이거나 '매우 가까움'으로 표현될 수 있어, 입력이 어떤 범위에 있는냐에 따라 현 상태에서 다른 상태로의 전이가 달라질 수 있다. 가장 적절한 출력을 위해 두 입력에 대한 조합을 만들고 그에 합당한 행동을 선택하여 전이가 이루어진다.

FuSM은 Hierarchical FSM보다 세밀하게 상태들을 나누고, 또한 범위를 두어 다양한 상태 전이가 가능하다. 경우에 따라서는 같은 입력에서도 다른 행동을 보여 줄 수 있다. 하지만 FuSM은 경우도 입력 범위에 따른 행동의 변화가 다양하지 못하다.

2.3 XML을 활용한 스크립트(Script)

스크립트(Script)는 소프트웨어에 실행시키는 처리절차를 문자(텍스트)로 기술한 것으로, 스크립트해석기를 서버 프로그램 안에 포함시킴으로써 가상 공간안의 객체 속성을 수정한다. 스크립트를 이용하므로 서버 프로그램을 고치지 않고 외부에서 수정할 수 있다. 이러한 스크립트의 사용은 편의성과 시간절약

에 장점이 있다.

XML은 데이터의 구조와 내용을 동시에 표현할 수 있는 마크업 언어로서, 데이터의 내용을 표현하는 공통 형식으로 그 응용분야가 다양하다. 이를 바탕으로 현재 업종별 데이터 형이나 비즈니스 규칙들을 XML로써 정의하고 있다. 대표적으로 전자상거래를 위한 ebXML, 신문업계의 NewsML 등의 각 비즈니스 영역에서뿐만 아니라, 수학적 표현을 위한 MathML, 휴대폰을 위한 WML, 음성 표현을 위한 VoiceML 등의 많은 업종 및 분야에서 XML을 기반으로 한 표준들이 만들어지고 있다. 또한 XML은 정보의 저장 및 표현이 용이한 장점이 있다. 기존의 스크립트들은 회사마다 새로운 프로젝트를 수행할 경우 새로운 스크립트를 만들었다. 그러나 XML을 이용하게 되면 표준화된 정보 표현방식으로 스크립트를 나타낼 수 있다.

또한 XML DB는 관계형 DB구조(Relational DB)가 아닌 OODB(Object-Orient DB)로서 트리구조를 가지기 때문에 검색에 유용하며 데이터 표현이 자유롭다. 그림 2는 RDB 구조와 OODB 구조의 차이점을 보여주고 있다.[9]

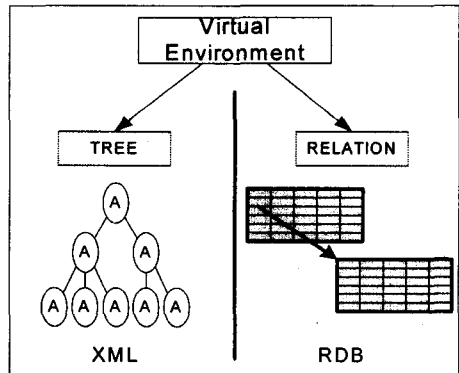


그림 2 OODB와 RDB의 데이터 표현 구조

3. 전체 시스템 구조

본 논문에서 제시하는 전체 시스템의 구조는 그림 3과 같다.

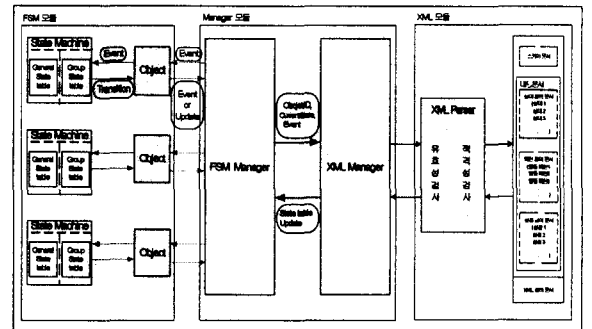


그림 3 전체 시스템 구조

3.1 FSM 모듈

FSM 모듈은 FSM Manager와 객체로 구성되며, 각 객체는 상태기계를 가지게 된다. FSM Manager는 가상 공간에서 일어나는 모든 이벤트(event)를 받아 객체나 XML모듈로 보낸다. 객체들은 FSM Manager로부터 이벤트를 상태기계(state machine)로 보내어 상태를 전이하게 된다.

FSM Manager는 이벤트가 들어오면 이벤트의 ObjectID를 확인하여 ObjectID에 맞게 객체에 전달한다. 이벤트를 받은 객체들은 자신들의 상태기계로 이벤트를 보내고, 상태기계는 상태기계 테이블을 기반으로 상태 전이를 하게 된다. 상태 전이로 발생하는 이벤트나 갱신(update)은 객체를 통해 FSM Manager

로 전달된다.

각 객체들이 가지는 상태기계(state machine)는 상태들을 테이블(table)형식으로 가지고 있으며, 이러한 상태 테이블의 행은 입력을 나타내고 열은 상태들을 나타낸다.

또한 하나의 상태기계는 일반 모드 상태기계 테이블(general state table)과 그룹 모드 상태기계 테이블(group state table)을 참조하여 모드 변경과 그룹 행동이 가능하다. 모드변경이란 하나의 행동 패턴에서 다른 행동 패턴으로 변경되는 것이며, 그룹 행동은 여러 객체가 함께 행동하는 Flocking 기법을 지칭한다. 모드 변경 이벤트는 경험치의 증가, 감소 또는 주변 객체와의 유대관계로 발생되며, 객체의 상태기계가 가지고 있던 일반 모드 상태 테이블을 갱신하게 된다. 상태 테이블의 갱신은 갱신 이벤트를 FSM Manager로 보내고 FSM Manager는 이 이벤트를 XML Manager로 보내어 이루어진다. 객체는 일반 모드 상태 테이블을 갱신한 후 상태를 천이 하게 된다.

또한 그룹 행동도 그룹 행동 이벤트가 발생하면 그룹 모드 상태기계 테이블을 이용하여 객체들이 그에 맞는 상태로 천이된다. 모드가 일반 모드에서 그룹 모드로의 변경은 동일한 GroupID를 가진 객체들만 변경한다. 이 이벤트가 발생하면 GroupID가 같은 객체에 그룹 이벤트를 보내고 객체들은 일반 모드에서 상태 천이의 변화인지, 아니면 일반모드에서 그룹모드로의 변경인지를 확인한다. GroupID가 같은 객체들은 일반 모드 상태기계테이블이 아닌 그룹 모드 상태기계 테이블을 가지고 행동을 천이 하게 되며, GroupID가 다른 객체들은 이 이벤트를 무시하게 된다.

그림 4는 이벤트가 전달되는 과정을 보여주는 다이어그램이다.

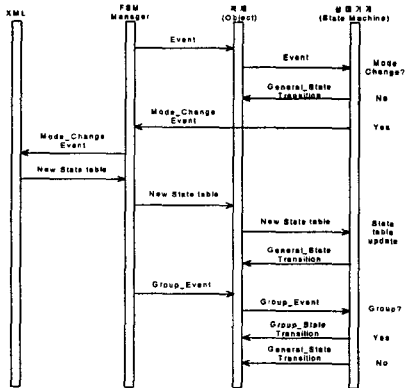


그림 4 이벤트 전달 과정

3.2 XML 모듈

XML 모듈은 XML Manager와 파싱 모듈, 그리고 XML 문서로 이루어진다. 여러 행동 패턴이나 상태들은 여러 문서에 정의하고, 파싱 모듈에서 적격성과 유효성을 검사한 후, XML Manager를 통해 FSM Manager로 전달된다. 모드변경 이벤트 발생시 XML Manager는 FSM Manager로부터 ObjectID, 현 상태정보 등을 받아 그에 맞는 상태정보나 행동 패턴을 찾아 FSM Manager로 전달한다. 상태 정의문서에는 상태들을 정의하고, 행동 패턴 정의 문서는 모드 변경이 이루어 질 수 있는 다양한 행동 패턴들을 정의한다.

XML Manager는 FSM Manager로부터 전달 받은 정보를 토대로 패턴 정의문서를 검색한다. 검색된 다른 행동 패턴의 정보는 파싱 모듈에서 유효성과 적격성을 검사하여 FSM Manager로 전송한다.

XML과 FSM 시스템을 상호 작용(interactive)하게 연결함으로써 행동 패턴들 간에 모드 변경이 가능하다. 모든 변경과 그룹

행동은 이벤트로 이루어지며, 모드 변경을 발생하게 하는 이벤트는 객체 주변의 환경변화나 객체의 경험치 또는 주변 객체와의 유대 관계 등을 모드 변경 이벤트로 할 수 있다. 그룹 행동은 적대 관계의 객체가 접근하거나, 공격을 할 때 그룹 행동 이벤트가 발생된다. 이러한 이벤트로 인하여 같은 입력 값에 대하여 다른 행동을 보여줄 수 있는 모드 변경과 그룹 행동이 가능하다.

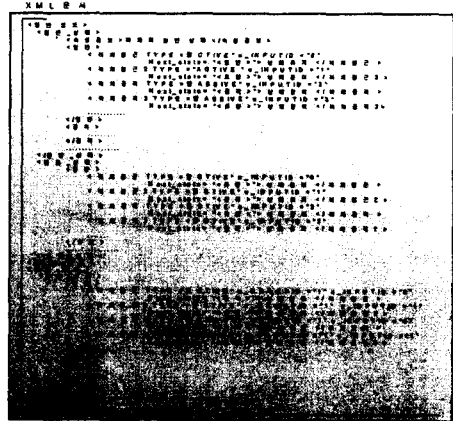


그림 5 XML 문서 구조

4. 결론 및 향후과제

본 논문에서 XML기반의 FSM 시스템을 제시하였다. 가상공간 내의 동적 객체들의 인공지능을 간단한 FSM을 이용하여, 모드 변경과 그룹 행동을 보여주었다. 이러한 XML을 활용한 FSM 시스템은 다양한 패턴의 정의는 물론, 객체의 상태 정의 및 수정, 확장이 용이하여, 다양한 응용 분야에서 활용될 수 있다. 향후 연구에서는 본 시스템을 보완하여 상태 정보를 정의한 상태 정의 문서의 적격성과 유효성을 검사하기 위한 태그들을 정의한다.

5. 참고문헌

- [1] 유석중외 2명, "공유가상공간에서 다중경로생물링을 이용한 동적 이벤트 필터링 기법", 정보과학회논문지, 제 26권, 11호, pp.1306-1313, 1999.
- [2] Sandeep Singhal, Michael Zyda, "Networked Virtual Environments", Addison-Wesley, 2001.
- [3] 박영식, "첫번째 사물 설계하는 기획자들을 위한 글", KGDC2002, 2002.
- [4] Andrew Luppnow, "Hierarchical AI", <http://www-cs-students.stanford.edu/~amitp/Articles/HierarchicalAI.html>
- [5] Steve Rabin외 47인 공저, "AI Game Programming WISDOM", pp.513-520, 정보문화사, 2003.
- [6] Mark Duloura외 공저, "Game Programming Gems", pp.299-332, 정보문화사, 2001.
- [7] 이경록외 2명, "온라인 컴퓨터 게임에서 지능형 NPC 구현을 위한 제어구조", 정보과학회 봄 학술발표논문집, Vol. 29, No. 1, pp.277-279, 2002.
- [8] 송병근외 3명, "FSM의 직관적인 모델링을 위한 에디터 설계", 정보과학회 2003년 춘계학술대회, Vol. 30, No. 1, 2003.
- [9] 조옥래, XML을 활용한 게임 스크립트, KGDC2002, 2002.
- [10] Microsoft, <http://msdn.microsoft.com/XML>
- [11] W3C, "MSXML 파서", <http://www.microsoft.com/korea/nsdn>
- [12] 김석중 역, "Microsoft XML 파서를 이용한 XML 프로그래밍", 인포북, 2003.