대규모 온라인 게임을 위한 XML 기반의 클라이언트-서버 메시지 정의 시스템

박학봉[O], 신지원, 오삼권
호서대학교 컴퓨터공학부
piaoxf@hanmail.net[O], yammi100@hotmail.com, ohsk@office.hoseo.ac.kr

The XML-based Client-Server Message Definition System for Massively Multi-player Online Games

Xuefeng Piao[O], Jiwon Shin, Samkweon Oh
Dept. of Computer Science, Hoseo University

요 약

MMORPG(Massively Multi-player Online Role Playing Games)와 같은 클라이언트-서버 기반의 대규모 온라인 게임은 정보 전송을 위한 많은 메시지들을 정의하여 사용한다. 그러나 게임의 기획이나 내용 및 기타 게임 요소들이 변경되면 개발자들은 메시지를 추가하거나 변경하는 문서를 작성해야 하고 작성된 문서에 근거하여 프로그램의 수정 작업을 수행해야 한다. 본 논문에서는 대규모 온라인 게임의 내용확장성으로 인해 지속적으로 반복되는 메시지의 추가 및 변경 작업을 용이하게 해주는 XML 기반의 메시지 정의 시스템을 제안한다. 제안된 시스템은 메시지 정의에 있어서 데이터의 구조 표현과 내용 표현이 동시에 가능한 XML을 이용했으므로 다양한 형식(format)의 메시지를 정의할 수 있으며, 대규모 온라인 게임뿐만 아니라 클라이언트-서버 기반의 일반 통신 응용 프로그램에도 사용할 수 있다.

1. 서론

1980년 보드(board)형 롤-플레잉(role-playing)기능과 채팅(chatting)을 결합한 MUD(Multi-User Dungeon) 게임으로부터 시작된 온라인 게임은 사용자가 네트워크를 통해 서버에 접속하여 진행하는 형태의 게임으로서, 1996년 MUD에 그래픽 출력 기능을 추가한 형태인 MUG(Multi-User Graphics) 게임이 상용화되면서 그 시장이 본격적으로 형성되기 시작했다[1]. 이러한 MUG게임은 지리적으로 분산된 수많은 사용자들이 인터넷을 통해 동시에 즐길 수 있는 MMORPG와 같은 대규모 온라인 게임으로 발전했다[2].

온라인 게임은 오프라인 게임과는 달리 상호작용성과 내용확장성을 가진다. 상호작용성은 클라이언트와 서버간에 정보가 양방향으로 전송되는 성질을 의미하고 내용확장성은 게임 내용이 지속적으로 변화하여 새로운 정보가 계속 추가되는 성질을 의미한다[1]. MMORPG와 같은 대규모 온라인 게임에서는 클라이언트-서버간의 상호작용과 게임 내용확장으로 인해 더 많은 메시지들의 지속적인 추가 및 변경이 필요하다. 이는 개발자들이 메시지를 추가하거나 변경하는 문서를 작성하고 작성된 문서에 근거하여 응용 프로그램을 수정해야 하는 반복된 작업을 요구한다.

1998년 W3C 권고안으로 발표된 XML(eXtensible Markup Language)은 SGML(Standard Generalized Markup Language)을 기반으로 한 마크업 언어로서 사용자가 임의로 태그를 정의 및 확장 할 수 있으며, 데이터의 내용 표현과 구조 표현이 동시에 가능하여 여러 분야에서 광범하게 활용되고 있다[3].

XML 스키마(schema)는 데이터가 마크업되는 방식으로서 XML 문서 안에서 사용되는 태그들의 속성과 구조, 그리고 어떤 값들을 가질 수 있는지에 대한 규정이다. 즉, 하나의 XML 문서 안에 들어가는 내용에 대한 규칙이다[4]. 그러므로 각 분야에서의 데이터 구조는 XML 스키마를 이용하여 정의할 수 있으며, 데이터 내용은 스키마에서 정의한 태그들을 이용하여 정의할 수 있다.

본 논문에서는 클라이언트-서버 기반의 대규모 온라인 게임에서 지속적으로 반복되는 메시지의 추가 및 변경 작업을 용이하게 하기 위해 XML 기반의 메시지 정의 시스템을 제안한다. 제안된 시스템은 메시지 정의 문서, 파싱 모듈, 유효성 검사 모듈의 세 부분으로 구성된다. 메시지 정의 문서는 메시지 형식을 정의하는 스키마 문서와 이 문서에 정의된 태그들을 사용하여 작성한 메시지 내용 문서로 구성된다. 파싱 모듈은 이 스키마 문서와 내용 문서를 입력으로 받아서 메시지의 구조를 정의하는 구조체 파일과 메시지 헤더(header)와 바디(body)의 정보들을 담고 있는 정보 파일들을 생성하며, 생성된 파일들은 라이브러리로 존재하는 유효성 검사 모듈 내에 포함된다. 응용 프로그램은 이 라이브러리를 사용하여 클라이언트-서버간에 송수신되는 메시지들의 유효성을 검사한다.

본 논문의 구성은 다음과 같다. 1장의 서론에 이어서 2장에서는 관련 연구에 대해 살펴보고, 3장에서는 클라이언트-서버 기반의 대규모 온라인 게임에서 일반 통신 시스템 모델을 설명하고 문제점을 기술한다. 그리고 4장에서는 XML 기반의 메시지 정의 시스템을 제시하고, 5장에서 결론을 맺는다.

2. 관련 연구

정보통신망의 구축과 응용산업 발전의 중요성이 대두되면서 온라인 게임은 중요한 핵심 산업으로 주목 받기 시작했다. 따라서 게임 그래픽, 게임 엔진 및 게임 서버 기술과 같은 관련 기술에 대한 연구가 세계적으로 광범위하게 진행되고 있다[5,6,7]. 그러나 대규모 온라인 게임에서 지속적인 게임 내용의 확장으로 인한 메시지 추가·변경의 반복적인 작업 문제를 해결하거나 완화하기 위한 연구는 찾아보기 힘들다.

XML은 다른 언어를 생성하기 위한 메타언어(meta-language)이다. 이를 바탕으로 현재 업종별 비즈니스 표준을 목표로 XML기반 표준화가 진행되고 있으며, 주로 데이터 형이나 비즈니스 규칙들을 XML로써 정의하고 있다. 대표적으로 전자상거래 표준안인 ebXML, 신문업계의 NewsML, 의료업계의 MML 등의 각 비즈니스 업종영역에서뿐만 아니라 수학식 표

현을 위한 MathML, 휴대폰을 위한 WML, 음성 표현을 위한 VoiceXML 등의 많은 업종 및 분야에서 XML을 기반으로 한 표준들이 만들어지고 있다[8,9].

3. 통신 시스템 모델

클라이언트-서버 기반의 통신 응용 프로그램에서 메시지의 송수신은 일반적으로 그림 1과 같이 이루어진다. 클라이언트에서 서버에 요청 메시지를 전송하면 서버는 수신된 메시지의 아이디, 길이 등의 값이 정의한 값과 일치하는지를 검사한다. 즉 메시지가 유효한지를 검사하고, 유효한 메시지이면 이에 대한 처리를 하고 결과 메시지를 클라이언트에 전송하며, 그렇지 않으면 오류 메시지를 전송한다. 이와 같은 통신 모듈은 다음 세 과정이 필요하다.

(1) 사용할 메시지들의 기능 분석을 통해 응용 프로그램 계층의 메시지 형식을 정의
(2) 메시지 형식에 맞게 메시지들을 정의
(3) 메시지 전송 모듈 및 수신된 메시지가 유효한지를 검사하는 유효성 검사 모듈 개발

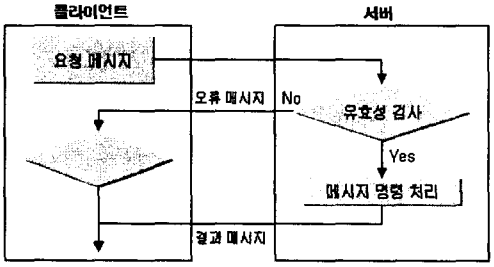

그림 1. 메시지의 송수신

그러나 대규모 온라인 게임에서는 게임 내용의 확장이나 기획의 변경으로 인해 지속적으로 메시지를 추가 및 변경해야 하므로 (2),(3)번의 작업을 계속하여 반복한다. 현재 이 작업은 개발자들이 프로그램을 수정하는 방법으로 진행되므로 오류 발생 확률이 높고, 많은 시간을 낭비할 수 있다. 또한 기존 응용 프로그램을 메시지가 추가·변경된 프로그램으로 갱신해야 하므로 안정적인 서비스를 제공하지 못할 수도 있다. 본 논문에서는 이와 같은 문제점들을 완화하기 위한 방법으로 XML을 이용한 메시지 정의 시스템을 제안하며, 제안된 시스템은 대규모 온라인 게임뿐만 아니라 일반 통신 응용 프로그램에도 모두 적용할 수 있다.

4. XML 기반의 메시지 정의 시스템

제안하는 시스템은 그림 2에서 보여주는 것과 같이 XML로 작성된 메시지 정의 문서, 메시지를 정의한 문서를 파싱하는 파싱 모듈, 그리고 클라이언트-서버간에 송수신되는 메시지가 유효한지를 검사하는 유효성 검사 모듈로 구성된다. 메시지 정의 문서는 파싱 모듈의 입력 데이터가 되며 출력으로 유효성 검사 모듈에서 사용되는 메시지의 구조를 표현하는 구조체 파일과 메시지의 정보를 담고 있는 정보파일들을 생성한다.

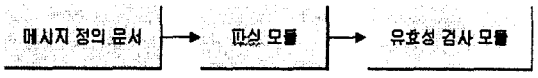

그림 2. 시스템 구성

4.1 메시지 정의 문서

메시지 정의 문서는 XML로 작성된 문서로서, 메시지 형식 정의 스키마 문서와 메시지 내용 정의 문서를 의미한다. 여기서, 메시지 내용 정의 문서는 형식 정의 스키마 문서에서 정의한 태그들을 사용하여 작성된다.

대규모 온라인 게임에서의 접속 메시지, 명령 메시지, 맵 데이터 갱신 메시지 등의 정보 전송 메시지들은 일반적으로 헤더와 바디로 구조를 가진다. 헤더는 각 메시지들에 공통적으로 들어가며 하나 이상의 필드(field)를 가질 수 있고 일반적으로 메시지를 식별하기 위한 필드와 길이를 나타내는 필드로 구성된다. 바디는 데이터의 내용을 담고 있는 필드들을 가지며 필드의 구조에 따라 나열 구조, 반복 구조, 혼합 구조, 바디가 없는 구조로 구분된다.

- ◆나열 구조: 하나 이상의 필드가 나열된 구조
- ◆반복 구조: 여러 개의 필드가 하나의 그룹을 구성하며 바디는 그룹으로 반복되는 구조
- ◆혼합 구조: 필드들의 나열과 필드 그룹의 반복이 혼합된 구조
- ◆바디가 없는 구조: 바디에 필드가 없는 구조

메시지 형식 정의 스키마 문서는 위와 같은 다양한 형식의 메시지들을 표현할 수 있는 태그들을 정의해야 한다. 따라서 본 논문에서는 메시지 내용 정의 문서에서 사용하는 태그들의 구조와 속성을 그림 3과 표 1과 같이 정의했다.

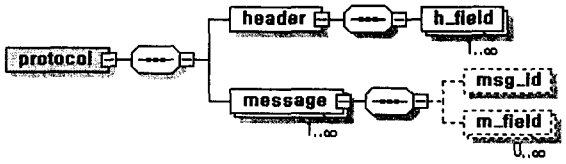

그림 3. 메시지 정의 태그 구조

표 1. 메시지 정의 태그 속성

태 그	속 성	데이터 형	설 명
protocol	version	string	메시지 정의 문서의 시작과 끝을 나타내는 태그 · version: 문서관리를 위한 버전 · desc: 문서에 대한 설명
	desc	string	
header	name	string	각 메시지에 공통적으로 들어가는 헤더를 정의하는 태그 · name: 헤더의 이름 · desc: 헤더에 대한 설명
	desc	string	
message	name	string	메시지의 시작과 끝을 나타내는 태그 · name: 메시지의 이름 · desc: 메시지에 대한 설명
	desc	string	
msg_id	name	string	각 메시지를 구별하는 아이디를 정의하는 태그 · name: 메시지 아이디 이름 · value: 메시지 아이디의 값 · desc: 메시지 아이디에 대한 설명
	value	int	
	desc	string	
h_field	type	data type	헤더의 각 필드를 정의하는 태그 · type: 필드의 데이터 타입 · size: 데이터 배열의 크기 · name: 필드의 이름 · desc: 필드에 대한 설명
	size	unsigned short	
	name	string	
	desc	string	
m_field	type	data type	메시지 바디 부분의 데이터 필드를 정의하는 태그 · type: 필드의 데이터 타입 · size: 데이터 배열의 크기 · name: 필드의 이름 · loop: 반복되는 바디 필드가 있는 경우 반복되는 바디 필드의 이름을 지정 · desc: 메시지 바디 필드에 대한 설명
	size	unsigned short	
	name	string	
	loop	message name	
	desc	string	

그림 4는 메시지 형식을 표현하는 스키마 문서에서 정의한 태그들을 사용하여 작성한 메시지 내용 정의 문서의 예이다.

```
<protocol xmlns="message" version="1.0" desc="Message Definition Document ">
    <header name="HEADER" desc="Protocol Message Header">
        <h_field type="WORD" name="wMsg" desc="Message ID Field"/>
        <h_field type="WORD" name="wSeq" desc="Packet Seq Number Field"/>
        <h_field type="WORD" name="wLength" desc="PDU Length Field"/>
    </header>
    <message name="CSP_REQ_LOGIN">
        <msg_id value="1001" desc="로그인 요청 메시지"/>
        <m_field type="char" size="18" name="szCharID" desc="Char ID Field"/>
        <m_field type="char" size="18" name="szPwd" desc="Password Field"/>
    </message>
    <message name="SCP_RESP_LOGIN">
        <msg_id value="9001" desc="로그인 응답 메시지"/>
        <m_field type="BYTE" name="byResult" desc="Result Field"/>
        <m_field type="int" name="nClientKey" desc="Client Key"/>
    </message>
</protocol>
```

그림 4. 메시지 내용 정의 문서의 예

4.2 파싱 모듈

파싱 모듈은 실행 프로그램으로 존재하며 본 시스템의 핵심 모듈이다. 입력 데이터는 메시지를 정의한 XML 문서가 되고 실행 결과로 유효성 검사 모듈에서 사용되는 메시지 정의에 맞는 구조체 파일과 정보 파일들을 생성한다. 이는 일반 XML 문서의 파싱과 파일 생성의 두 과정을 거치게 된다. 파싱은 마이크로소프트사에서 제공하는 MSXML파서를 사용하여 문서의 적격성(well-formedness)검사와 의미 유효성(semantics validation) 검사를 하고, 파일 생성은 DOM(Document Object Model) API를 이용하여 메시지 정의에 맞는 구조체 파일과 메시지의 아이디나 길이 등의 정보를 담고 있는 정보 파일들을 생성한다[10,11]. 그림 5는 실행결과의 예이다.

Structure File

```
#ifndef __SOURCECODE_H__
#define __SOURCECODE_H__
// Version 1.0
// Structure 문서
#define REQ_SERVERLIST   0x10

typedef struct _S_HEADER
{
    char    cMsgID;
    shrot   sLength;
}S_HEADER;

typedef struct _S_SERVER_INFO
{
    int     nSvrID;
    char    szSvrIP[16];
    UINT    uSvrPort;
}S_SERVER_INFO;

typedef struct _S_SERVER_LIST
{
    int     nServerNum;
    S_SERVER_INFO* ServerInfo;
}S_SERVER_LIST;

#endif
```

Header Info File

```
Header 정보 문서 1.0

0x10 3 4 24
```

Message Info File

```
Message Info 문서 1.0

HEADER    1 2 1 2
SERVER_INFO 2 4 4 1 1 5
SERVER_LIST  3 2 4 s2
```

그림 5. 파싱 결과 예

4.3 유효성 검사 모듈

유효성 검사 모듈은 파싱 모듈에서 생성된 파일들에 근거하여 클라이언트-서버간에 송수신되는 메시지들이 유효한지를 검사한다. 이 모듈은 그림 6과 같이 선행 처리기 #define 지정어를 사용하여 파싱 모듈에서 생성된 파일들을 포함한 라이브러리 형태 존재하며 외부호출 함수들을 제공한다. 응용 프로그램은 이 함수들의 호출을 통해 수신된 메시지의 아이디나 길이 등의 값들을 유효성 검사 모듈에 인자로 넘겨주면 유효성 검사 모듈은 이 값을 정보 파일에 저장된 값과 비교하여 정의한 값과 일치하는지를 검사한다. 그리고 그 결과를 응용 프로그램에게 반환한다. 그러므로 메시지의 추가 및 변경은 새로 추가·변경한 메시지들을 포함한 구조체 파일과 정보 파일들의 갱신만으로도 이루어진다.

```
#define     STRUCT_FILENAME     "struct.h"
#define     MSGINFO_FILENAME    "msginfo.inf"
#define     HDINFO_FILENAME     "hdinfo.inf"
```

그림 6. 구조체 파일과 정보 파일의 선행처리 예

5. 결론 및 향후 연구

대규모 온라인 게임에서 지속적으로 반복되는 메시지 추가 및 변경은 많은 시간의 낭비를 초래할 수 있고, 응용 프로그램의 오류 발생 확률도 높다. 또한 안정적인 서비스를 제공하지 못할 수도 있다. 본 논문에서는 이런 문제점을 완화하고 메시지의 추가 및 변경을 용이하게 하기 위해 데이터의 구조와 내용이 분리된 XML를 이용하여 메시지를 정의하는 XML 기반의 메시지 정의 시스템을 제안했다. 이 시스템을 사용하면 개발자들은 기존 메시지 정의 문서를 수정하고, 파싱 모듈로써 파싱하여 그 결과로 생성된 구조체 파일과 정보 파일들로 기존 파일들을 교체하는 것으로 메시지 추가 및 변경을 할 수 있다.

본 논문에서 제안하는 XML 기반의 메시지 정의 시스템은 대규모 온라인 게임뿐만 아니라 일반 클라이언트-서버 응용분야에서도 사용될 수 있는 범용 시스템이며, 데이터의 구조와 내용이 분리된 XML의 장점을 이용했으므로 다양한 형식의 메시지 정의도 쉽게 할 수 있다.

향후 연구에서는 본 시스템을 보완하여 메시지의 아이디나 길이 등의 정보뿐만 아니라 메시지의 시퀀스(sequence) 검사도 가능하고, LAN 환경뿐만 아니라 이동 통신환경에서도 사용 가능한 메시지 정의 시스템에 연구할 계획이다.

참고문헌

[1] 한국 게임산업개발원, 게임백서, pp.564-574

[2] 최석우, "리니지 개발 사례", 정보처리학회지 제9권 제3호, 2002. 5

[3] 김창수, 정희경, "XML 응용 개발환경", 정보과학회지 제19권 제1호, 2001. 1

[4] 황병연, 김연, "XML 스키마 발전 동향", 정보처리학회 제8권 제3호, 2001. 5

[5] 이남재, 곽훈성, "완전한 3차원을 지원하는 온라인 RPG를 위한 맵 관리 방법", 정보처리학회 논문지 B, 제9권, 제6호, 2002

[6] J. Smed, T. Kaukoranta, and H. Hakonen. "Aspects of networking in multiplayer computer games", In International Conference on Application and Development of Computer Games in the 21st Century, Hong Kong, November 2001.

[7] 신동원, "The Study of Operating System for High-Tech Network Game"

[8] 최영근, 정계동 공역, "XML How To Program", Prentice Hall, 2001

[9] 이강찬, 손홍, 박기식, "XML 표준화 동향", 정보과학회지, 제19권, 제1호

[10] "msxml파서",http://www.microsoft.com/korea/msdn

[11] 선승상, 박상윤, 엄영익, "XML 문서의 객체지향적 관리를 위한 XML DOM 소프트웨어의 설계 및 구현", 정보처리학회 춘계학술대회, Vol 07, No 1, 2000.4