

KVID기반 KVMF메시지의 동적 메시지 처리 연구

정회원 조철영^{*o}, 이준표^{*}, 권철희^{*}, 조한준^{*}

A Study on Dynamic Message Processing for KVMF Messages based on KVID

Chul-young Cho^{*o}, Jun-pyo Lee^{*}, Cheol-hee Kwon^{*}, Han-jun Cho^{*} *Regular Members*

요 약

무기체계간의 상호 운용성 증대를 위하여 한국형 가변 메시지 포맷 (Korean Variable Message Format: KVMF)을 적용 시 메시지 구조의 저장 위치와, 파서의 구조, 파서코드 생성기 설계 방법을 고려하여야 한다. 특히 기존 무기체계에 KVMF를 적용하는 경우를 위하여 성능 및 유지보수성 등을 분석하여 메시지 파서를 설계하는 것이 중요하다. 본 논문에서는 KVMF 통합 데이터베이스 (KVMF Integrated Database: KVID)를 기반으로 기존 도입된 장비와 차기 도입될 장비 모두에 적용할 수 있는 실시간 동적 KVMF메시지 파서 설계 방법을 제안하고 제안한 방법을 포병대사격지휘체계에 적용하여 메시지 변환의 정확성을 확인하도록 한다.

Key Words : KVMF, KVID, 상호운용성, 데이터링크, 동적파서

ABSTRACT

It is important to consider saving location of message structures, structure of parser and creating method of parser generator for adopting KVMF to increase interoperability among different weapon systems. Especially, it requires to design the message parser by analyzing performance and maintenance when we adopting KVMF to previously developed weapon systems. In this paper, we propose the method of concurrent dynamic message processing for KVMF message based on KVID to support the developed and next-developed weapon systems. We show message conversion accuracy and performance effectiveness by adopting proposed method to Battalion Tactical Combat System.

I. 서 론

네트워크 중심전 (Network Centric Warfare; NCW)은 육·해·공군의 전력을 유기적으로 통합하여 효과적으로 운용할 수 있게 하는 현대 전투체계의 핵심이라 할 수 있다. 이를 위하여 각 전투체계 간의 상호운용성은 매우 중요하다. 그러나 현재 한국군의 무기체계는 다양한 전술데이터링크를 사용함에

따라 상호 무기체계간의 원활한 연동 및 상호운용이 어렵다^[1]. 최근 들어 다양한 형태의 통합전술데이터링크가 개발되고 있지만 이미 개발이 완료되어 사용되고 있는 무기체계에 이를 적용하기는 추가적인 비용과 개발인력이 필요하여 이에 대한 보완이 시급하다. 한국군은 NCW를 대비하여 한국형 가변 메시지 포맷 (Korean Variable Message Format; KVMF) 전술데이터 링크 기술 확보를 위하여 2004

* LIG넥스원 연구개발본부 소프트웨어 연구센터(chulyoungcho@lignex1.com), (° : 교신저자)

논문번호 : KICS2011-11-538, 접수일자 : 2011년 12월 21일, 최종논문접수일자 : 2011년 2월 13일

년부터 연구 개발을 추진 중이다. 연구 개발 계획에는 메시지 포맷 설계, 메시지 처리기, 프로토콜 설계, 검증 도구 개발 등이 포함되어 있다^{2,3}. KVMF는 상호운용성 증대를 위하여 메시지 구조, 처리 규칙, 자료 항목들을 저장하는 KVMF 통합 데이터베이스(KVMF Integrated Database; KVID)를 활용하도록 설계되었다. 기존에 도입된 장비의 메시지 포맷을 KVMF 형태로 변경하기 위해서는 비용, 시간, 일부 기술적 제약이 발생하여 비효율적이다. 그러므로 현재 연구 개발 중인 메시지 처리기는 단순히 KVMF 메시지만을 해석하는 것이 아니라 기존에 도입된 장비의 메시지 포맷을 KVMF 형태로 상호연동 가능하도록 하여야 한다. 이는 각 장비에 특화된 메시지 포맷을 KVMF로 변환하는 데이터 변환기와 메시지 파서를 개발 하는 것으로 요약 할 수 있다. 메시지 처리기는 KVID를 활용하여 KVMF 메시지에 공통으로 적용되는 문법과 메시지 구조의 위치, 파서와 메시지 간의 관계를 어떤 구조로 할 것인지에 처리 성능 및 유지보수성이 결정된다.

기존에는 KVMF 형태의 메시지 포맷을 사용하지 않는 무기체계는 KVMF 사용을 위하여 개발자가 직접 메시지 생성 및 송수신에 대한 모든 소스코드를 변환하거나 KVID에서 KVMF 문법을 파일로 추출하여 각각의 메시지에 맞는 메시지변환 프로그램을 별도로 만들어 사용하였다. 이에 따라 추가적인 개발시간과 인력이 필요하였다. 또한, 메시지에 변경이 필요한 경우 재작업을 수행해야 하는 단점이 있었다. 미군의 가변형 메시지 포맷(Variable Message Format; VMF)에는 XML-VMF 포맷이라는 표준이 존재한다. 이 표준의 목적은 언제 어디서나 데이터의 확인 및 사용을 위하여 모든 데이터에 메타데이터를 태깅하여 사용하도록 하는 것이다. 이를 활용하여 기존의 무기체계의 데이터를 XML-VMF 표준 스키마를 활용하여 변환이 가능하나 기존데이터의 변환시간은 물론 파일 입출력과 XML 파싱시간에 따른 실시간적인 데이터 처리가 불가능 하여 빠른 의사결정이 필요한 무기체계 시스템에는 비효율적이다.

본 논문에서는 국방과학연구소에서 개발한 KVID 1.0을 기반으로 기존 도입 장비와 차기 도입 될 장비에 범용적으로 적용 가능한 KVMF 메시지 처리기 설계 방법을 제안한다. 메시지 처리기는 인코딩, 디코딩을 모두 수행 할 수 있어야 하며, KVMF 메시지는 물론 다른 장비에서 사용되는 특정 형식의 메시지를 모두 수용 가능하여야 한다. 송수신 시 각

메시지는 KVID와 각 시스템 고유 메시지 포맷에 따라 메시지 변환 문법을 생성하고 생성된 문법을 이용하여 실시간으로 동적 파서를 생성하여 생성된 문법을 이용하여 디코딩과 인코딩을 수행하도록 한다. 이러한 처리과정을 통하여 문자형 메시지를 비트 단위 메시지로 비트 단위 메시지를 문자형 메시지로 상호 변환 시킨다. 마지막으로 제안한 방법을 현재 육군에서 운용중인 포병 사격지휘체계(Battalion Tactical Command System; BTCS)와 KVMF 메시지를 검증하는데 사용하는 KVMF 메시지 검증도구(KVMF Message Test Tool; KVTT)에 적용하여 타당성을 검증하도록 한다.

II. 본 문

KVMF 메시지 처리기를 설계하는데 있어 기존 도입된 장비를 수용하기 위하여 다음 두 가지 요건을 만족하여야 한다. 첫째, 상호 데이터 변환의 정확성이다. 모든 시스템은 고유한 데이터 표현 방법을 가지고 있다. 단위, 부호, 수의 범위 등 상이한 두 시스템 사이의 데이터를 정확히 변환하는 것은 데이터 전달의 정확성을 위하여 매우 중요하다. 둘째, 상호시스템 간 서로 다른 메시지 포맷의 변환이다. 각 시스템은 고유의 메시지 포맷을 KVMF 메시지 형태로 변환하기 위하여 상호 시스템의 고유 문법에 따라 포맷을 변환하여야 한다. 이는 시스템의 고유 메시지 구조를 어떻게 표현하고 어디에 저장 하는지에 따라 메시지 처리 시의 성능과 향후 메시지 추가 및 변경에 따른 유지보수성이 달라지기 때문이다.

KVMF 동적 파서 생성기는 그림 1과 같은 구조로 구성된다. BTCS 메시지와 KVMF 메시지를 상호변환하기 위하여 어댑터 모듈을 통해 메시지의 종류와 스키마 저장 여부 등을 확인한다. 데이터 추출기는 메시지 변환 전 KVID를 기반으로 변환하고자 하는 데이터 값을 추출하는 역할을 한다. 메시지 문법 저장소에서 변환 전 메시지의 데이터 필드와 KVMF 메시지 데이터 필드 간 의미매핑을 통해 데이터 수집을 수행한다. 동적 파서 생성기는 비트 단위 값을 문자형 값으로 변경 및 그 반대의 연산을 수행한 후 메시지에 따라 문법을 통해 동적 메시지 파서가 생성되어 해당 메시지를 변환하고자 하는 파싱트리를 생성하고 생성된 메시지의 파싱트리의 데이터 필드에 값을 삽입한다. 이때, KVID를 통하여 KVMF 메시지의 스키마를 읽어오고 해당 스키

마에 따라 파서 생성 문법을 기반으로 파싱트리를 구성하여 메시지 변환 작업을 수행하도록 한다.

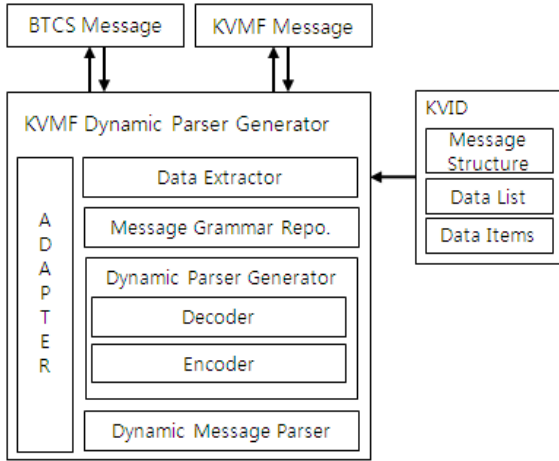


그림 1. 제안하는 KVMF 동적 파서 생성기 구성도
Fig. 1. Configuration of KVMF dynamic parser generator

우선 기존 메시지를 KVMF 메시지로 변환할 때 가장 중요한 것은 데이터 값의 올바른 변환이다. 이를 위하여 KVMF 메시지의 포맷과 데이터 항목을 저장하고 있는 KVID를 다음과 같은 항목에 따라 분석하여 데이터 변환모델을 확립한다. KVID에서 값은 비트코드로 표현되는 항목 값 (Data Item ;DI) 으로 구성된다. 이 DI값은 다음의 3가지 항목으로 분류된다. 첫째, 문자열로 구성되어 있는 항목으로 특정 비트 값과 데이터 항목이 일대일로 매핑 된다. 예를 들면 비트 1은 수면, 비트2는 수면 밑, 비트 3은 육지, 비트 4는 공중과 같이 표현된다. 둘째, 항목 값이 숫자로 표현되는 데이터이다. 주로 거리, 온도, 각도 및 시간 등이 이 표현 범주에 속한다. 이 항목 값은 KVID에서 ‘multiplier field’를 사용하여 데이터의 최소, 최대값을 표현한다. 또한, 음수

값 표현을 위하여 2의 보수를 사용한다. 셋째, ‘NO STATEMENT’, ‘UNKNOWN’, ‘PENDING’, ‘ILLEGAL’, ‘UNDEFINED’와 같이 정의 되지 않은 필드 혹은 불법문자 등의 미사용, 사용 금지 범위 비트로 구성 된다. KVID에서 정의 된 위와 같은 항목 값과 각 시스템에서 사용하는 데이터 값을 매핑하여 변환하기 위하여 우선 기존 시스템에서 사용하는 데이터 유형을 분석하고 정리하여야 한다. 정리 된 유형들을 KVID의 항목 값에 맞도록 변환하는 것은 정확한 데이터를 단일 메시지로 전송 하는데 있어 가장 중요한 요소이다.

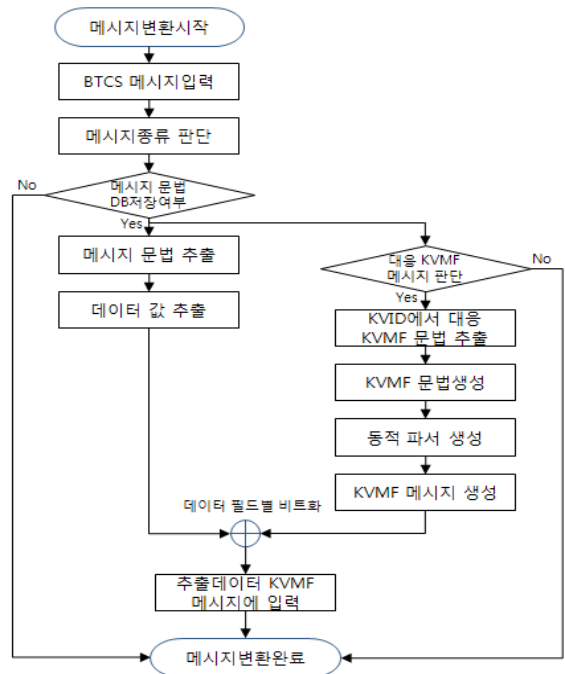


그림 2. 제안하는 메시지 변환 순서도
Fig. 2. Flow chart of message transformation

기존 시스템의 데이터를 정제한 후 그림 2와 같

인덱스	자료항목(그룹(OF))	자료항목(DU)	자료항목(DU) 한글명	자료항목(DU) 영문명	비트수	범례	그룹코드	반복그룹코드	반복수	정확도 설명
1.	4057	2	사격임무 유형	FIRE MISSION TYPE	4	M			0	
2.	4014	2	필드 존재 지시자	FPI	1	M			0	
2.1	4003	1	표적 번호	TARGET NUMBER	28	X			0	
3.	4014	1	그룹 존재 지시자	GPI	1	M			0	GPI FOR G1. TARGET DATA
3.1	4079	7	포복선 지시자	GUN-TARGET LINE INDICATOR	1		G1		0	
3.2	4014	2	필드 존재 지시자	FPI	1		G1		0	
3.2.1	4028	1	관측(觀自) 방위각	OBSERVER-TARGET AZIMUTH	13		G1		0	
3.3	4014	1	그룹 존재 지시자	GPI	1		G1		0	GPI FOR G2. TARGET LOCATION.
3.3.1.1	4045	1	그룹 반복 지시자	GRI	1		G1G2	R1	15	GRI FOR R1
3.3.1.2	4014	2	필드 존재 지시자	FPI	1		G1G2	R1	0	
3.3.1.2.1	4085	99	표적 성분 번호	TARGET ELEMENT NUMBER	4		G1G2	R1	0	
3.3.1.3	281	407	표적 위도	TARGET LATITUDE	25	X	G1G2	R1	0	
3.3.1.4	282	407	표적 경도	TARGET LONGITUDE	26	X	G1G2	R1	0	
3.3.1.5	4014	2	필드 존재 지시자	FPI	1		G1G2	R1	0	
3.3.1.5.1	4130	4	표적 고도	TARGET ELEVATION, MSL	17	X	G1G2	R1	0	GPI FOR G3. FIXED POINT DATA.
3.3.1.6	4014	2	필드 존재 지시자	FPI	1		G1G2	R1	0	
3.3.1.6.1	4214	2	표적 고도 HAE	TARGET ELEVATION, HAE	22	X	G1G2	R1	0	
3.3.1.7	4014	2	필드 존재 지시자	FPI	1		G1G2	R1	0	
3.3.1.7.1	4119	1	표적 위치 정확도	TARGET LOCATION ACCURACY	10		G1G2	R1	0	GPI FOR G4. MOVING TARGET DATA.
3.3.1.8	4014	1	그룹 존재 지시자	GPI	1		G1G2	R1	0	
3.3.1.8.1	4025	1	표적 일반유형	TARGET GENERIC TYPE	5	X	G1G2/G3	R1	0	
3.3.1.8.2	4026	1	표적 상세유형	TARGET SUBTYPE	8	X	G1G2/G3	R1	0	
3.3.1.8.3	4014	2	필드 존재 지시자	FPI	1		G1G2/G3	R1	0	
3.3.1.8.3.1	4027	1	방호 정도	DEGREE OF PROTECTION	4	X	G1G2/G3	R1	0	

그림 3. KVID에 저장되어 있는 K02.4 사격요청 메시지 구조 일부
Fig. 3. Sample message structure of K02.4 fire request in KVID

은 순서로 메시지 변환을 수행한다. 변환하고자하는 메시지가 입력되면 해당 메시지의 종류를 판단하고 해당 메시지에 대한 구조데이터가 존재하는지 여부를 판단한다. 해당 시스템의 문법이 존재하는 경우 메시지에서부터 데이터를 추출한다. 동시에 해당 메시지에 대응하는 KVMF 메시지의 문법구조를 KVID에서 추출하고 동적으로 문법에 따라 파서를 생성하여 KVMF 메시지 파싱트리를 구성한다. 생성된 파싱트리에 추출한 데이터를 비트화하여 적절한 대응 필드에삽입하여 메시지 변환을 완료 한다. 그림 2의 메시지 변환 순서에 따라 동적으로 변환하고자하는 메시지에 대응되는 KVMF 메시지로 변환하는 동적 파서를 생성하고 데이터를 입력하는 방법에 대하여 구체적으로 설명하도록 한다. 그림 3은 KVID에 저장되어있는 K02.4 메시지(사격요청) 기술서의 일부분을 나타낸다^[5]. 그림 3의 인덱스 항목을 통한 계층화된 데이터 구조 표현과 자료항목그룹(DFI), 자료항목(DUI) 그리고 각 필드의 비트 수와 그룹코드, 반복 그룹코드, 반복수, 정확도 설명 등을 통하여 메시지의 구조를 나타내고 이를 이용하여 메시지 생성문법을 작성 하도록 한다^[4].

표 1. 제안하는 메시지 생성문법
Table. 1. Message Generating Grammar

<pre> Message ::= SFIELD SFPI SGPI SFRI Message SGPI ::= "{" SFIELD SGPI SFPI SGRI SFRI "}" SGRI ::= "(" [0-9]+)"{" SFPI SGPI SGRI SFRI SFIELD "}" SFRI ::= "(" [0-9]+)"{" SFIELD "}" SFPI ::= SFIELD SFRI SFIELD ::= "FIELD" SFPI ::= "FPI" SGPI ::= "GPI" SFRI ::= "FRI" SGRI ::= "GRI" CFIELD ::= SFIELD (" [0-9]+)"{" [0-9]+)"{" [0-9]+)"";" </pre>

KVID에 저장되어 있는 포맷에 따라 동적 메시지 생성 문법 규칙을 다음과 같이 제안한다.

- 규칙 1 : Atomic 단위인 필드는 'FIELD' token 다음에 필드 사이즈를 괄호 '(' 안에 표시한

다. 또한, DFI/DUI 값을 '[' 안에 표시한다.

- 규칙 2 : 존재 지시자는 1비트로 필드 사이즈를 표기하지 않는다.

- 규칙 3 : 그룹 반복 인덱스(Group Repeat Index ;GRI), 필드 반복 인덱스(Field Repeat Index ;FRI)는 최대 반복수를 '(' 안에 표시한다.

- 규칙 4 : 그룹 존재 인덱스(Group Presence Index ;GPI), GRI, FRI는 '{' 으로 KVID의 index 필드를 이용하여 계층을 구분한다.

위와 같은 규칙에 따라 메시지 생성 문법을 EBNF형식으로 나타내면 표 1과 같다. 표 1의 생성 규칙에 따라 생성된 K02.4 메시지의 메시지 생성 문법은 다음과 같다.

```

FIELD(4) [4057][2];
FPI FIELD(28) [4003][1];
GPI{
    FIELD(1) [4079][7];
    FPI FIELD(13) [4028][1];
    GPI{
        GRI R1(15) {
            FPI FIELD(4) [4085][99];
            FIELD(25) [281][407];
            FIELD(26) [282][407];
            FPI FIELD(17) [4130][4];
            FPI FIELD(22) [4214][2];
            FPI FIELD(10) [4119][1];
            GPI {
                FIELD(5) [4025][1];
                FIELD(8) [4026][1];
                FPI FIELD(4) [4027][1];
                생략
            }
        }
    }
}
GPI {
    FPI FRI(3){
        FIELD(4) [4115][2];
    }
}
GPI {
    생략
}
                
```

다음으로 작성 된 문법을 이용하여 추상 구문 트리를 구성하도록 한다. 그림 4는 표 1의 생성 문법을 이용하여 구성한 파싱트리의 일부분을 나타낸다

[4]. 데이터 값을 저장하는 EData 노드는 DFI와 DUI 값과 데이터 값을 저장한다. 위와 같이 생성된 파싱 트리는 한 개의 완벽한 메시지를 나타낸다.

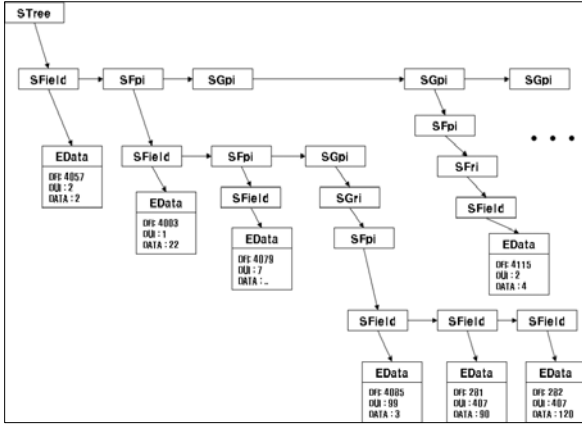


그림 4. K02.4 사격요청 메시지 파싱트리 예
Fig. 4. Example of message parsing tree of K02.4

또한, 송/수신 시 파싱트리를 이용하여 상위 혹은 하위 레이어로 메시지를 전송 할 때에 트리를 순회 하면서 비트의 문자열 화 혹은 문자열의 비트 화를 쉽게 수행 할 수 있다.

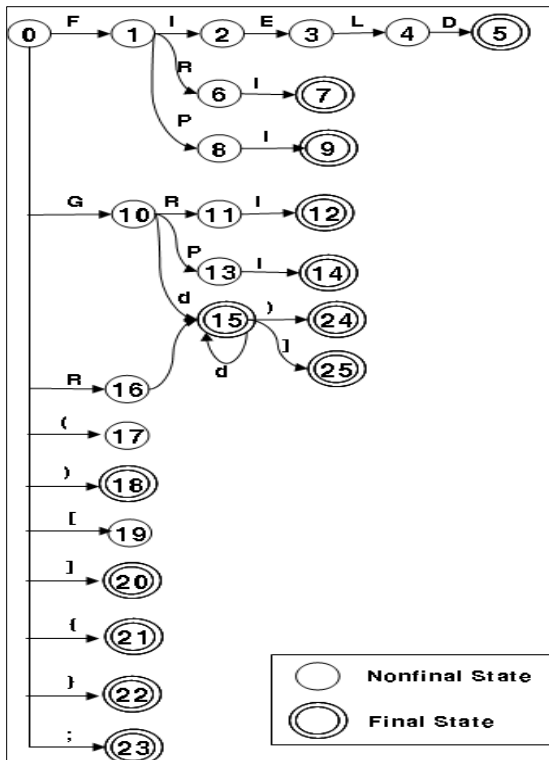


그림 5. 제안하는 파싱 트리 생성 Finite Automata
Fig. 5. Finite automata of parsing tree generating

파싱트리는 계층형 트리구조로써 GPI, GRI, FRI, FPI, FIELD는 그림 5의 파싱 트리 생성 FA(Finite Automata)에서 Final state로 state node를 구성한다.

데이터 값을 저장하는 EData node는 DFI와 DUI 값과 데이터 값을 저장한다.

이와 같이 추상 구분 트리를 구성함에 따라 KVMF의 케이스 및 컨디션을 효율적으로 체크할 수 있으며 데이터 변환과 메시지 파싱을 병렬적으로 수행함으로써 성능 향상을 기대 할 수 있다[8].

```

- <KVMFMessage_K02.40>
- <CASE>
- <Case_1>
- <LOGIC VAL="AND">
  <FIELD VAL="4057_2">0</FIELD>
  <FIELD VAL="R1">0</FIELD>
  <FIELD VAL="G6">0</FIELD>
  <FIELD VAL="G7">0</FIELD>
  <FIELD VAL="G9">0</FIELD>
</LOGIC>
</Case_1>
- <Case_2>
- <LOGIC VAL="AND">
  <FIELD VAL="4057_2">1</FIELD>
  <FIELD VAL="4003_001">1</FIELD>
  <FIELD VAL="G1">0</FIELD>
</LOGIC>
</Case_2>
+ <Case_3>
</CASE>
+ <CONDITION>
</KVMFMessage_K02.40>
    
```

그림 6. 제안하는 K02.4 CASE 정의 XML 구조
Fig. 6. XML structure of K02.4 case

```

- <KVMFMessage_K02.40>
+ <CASE>
- <CONDITION>
- <COND_1>
- <IF>
- <LOGIC>
  <FIELD VAL="G1">1</FIELD>
</LOGIC>
</IF>
- <THEN>
- <LOGIC>
  <FIELD VAL="4031_1">0</FIELD>
</LOGIC>
</THEN>
</COND_1>
- <COND_2>
+ <IF>
+ <THEN>
</COND_2>
+ <COND_3>
</CONDITION>
</KVMFMessage_K02.40>
    
```

그림 7. 제안하는 K02.4 Condition 정의 XML 구조
Fig. 7. XML structure of K02.4 condition

구성된 파싱트리에 의미론적 규칙 적용을 위한 케이스와 컨디션 적용 방법은 다음과 같다. KVMF에서 케이스는 하나의 기능 단위 전문으로 분류 및 정의할 수 있다. 이 때, 주의 할 점은 하나의 케이스는 여러 개의 전문으로 구성 가능하다는 것이다. 본 연구에서는 케이스와 컨디션 정보는 XML 형식으로 새로 정의하여 그림 6, 7과 같은 형식으로 저장하였다^{15,6)}. 각 노드의 설명은 다음과 같다. <CASE_X>는 각 전장영역 메시지의 케이스 구성을 나타내고 <LOGIC>노드는 <AND>, <OR>, <XOR>등으로 구성된 논리 연산을 기술한다. 이미 생성된 파싱트리를 순회하면서 이접정규형(Disjunctive Normal Form ;DNF)으로 케이스를 변환하여 파싱트리의 무결성을 검증한다. 컨디션을 기술하는 <CONDITION>노드는 <IF>, <THEN>같은 조건문 형식으로 각 메시지의 무결성을 보장하기 위한 정보를 저장한다. 케이스와 마찬가지로 불린형식으로 표현하여 DNF로 변환하여 메시지를 검증하도록 한다. 위와 같은 일련의 흐름을 거쳐 KVMF메시지로 변환이 되어 전송을 하게 된다. 메시지를 수신 받아 디코딩 시에는 역으로 수행하여 각 시스템에 맞는 메시지 형식으로 역 변환을 하여 전송함으로써 메시지 상호 교환을 원활히 이루어지도록 한다.

III. 실험 방법 및 결과

3.1. 실험환경 및 방법

본 논문에서는 제안하는 방법에 따라 포병대대사격지휘체계 시스템에 KVMF를 적용하여 검증을 수행하였다. 실험 환경은 그림 8과 같이 두 대의 전술통제기를 무선으로 연결하고 중간에 KVTT 소프트웨어⁷⁾를 활용하여 송수신 메시지를 캡처하여 메시지 변환의 정확성 및 무결성을 확인하도록 한다. 또한, 메시지 변환 시 성능 측정을 위하여 메시지를

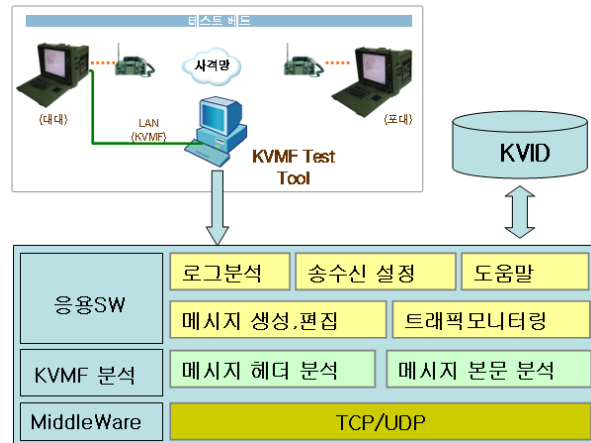


그림 8. 제안하는 KVTT활용 실험환경 구성도
Fig. 8. Configuration of test-bed based on KVTT

수신 한 시점부터 데이터 변환 모듈의 연산이 끝나는 시점까지의 시간을 메시지의 크기에 따라 성능을 측정하여 분석하도록 한다. 이때, KVMF 스키마를 저장하고 있는 KVID의 저장위치에 따른 성능차이를 평가하기 위하여 MS-SQL 2008 Server와 메인메모리 데이터베이스를 사용한다. 미군 VMF 표준에 있는 XML-VMF를 활용하여 기존 메시지를 XML 메시지 파일로 변환하는 시간을 측정하여 성능을 비교하도록 한다.

BTCS에 KVMF를 적용하기 위하여 BTCS에서 사용되는 메시지의 데이터 유형을 정리하여 데이터 변환 모델을 표 2와 같이 작성하였다. 각 데이터 유형에 따라 추가적으로 필요한 정보를 정리하고 그에 따라 변환 방법을 데이터베이스에 저장 혹은 변환 함수를 이용하여 데이터를 변환하는 방법을 사용하였다. 이를 통하여 새로운 장비와 KVMF를 연동 시 데이터 변환 모듈의 데이터베이스화만으로 쉽게 변환 가능하도록 하였다.

3.2. 실험 결과

그림 9는 테스트베드에서 송수신 하는 메시지를

표 2. BTCS to KVMF 데이터 변환 필드 데이터 유형
Table. 2. Type of data converting fields for BTCS to KVMF messages

유형	형태 예	필요사항	변환방법
1:1매칭	1/2 - 사격/비사격	DI Set의 해당번호	데이터베이스 지정
부호+숫자	[R/L]+XXXX(X:숫자)	부호 및 부호위치	데이터베이스 지정
숫자	일반 아라비아 숫자	단위(최소배율)	변환함수
문자+숫자	FP10, FF11 등	비트 그룹 구성 포맷	데이터베이스 지정
음어	-	7-8bit padding	변환 함수
주해	일반 문자열	한글/영문(비트수)	by-pass
URN	군 주소체계	-	변환 함수

수신 받아 KVMF 메시지 구성의 정확성을 KVTT를 활용하여 판단한 화면을 나타낸다. 본 실험에서는 포병대대사격지휘체계의 KVMF 적용 메시지인 포대이상 전문 92개와 포대이하 전문 27개, 총 119개 전문에 대하여 실험을 수행하였으며 모든 데이터가 정확하게 변환됨을 확인하였다.

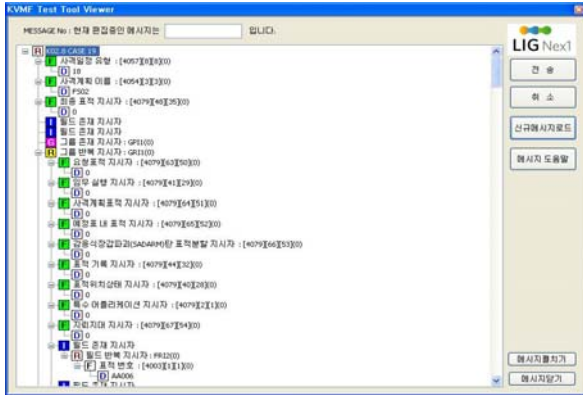


그림 9. KVTT 활용 메시지 변환 검증 화면
Fig. 9. GUI interface of message converting verification based on KVTT

그림 10은 제안한 KVMF 동적 메시지 처리방법을 XML-VMF 형태로 변환하는 방법과 비교 평가한 실험 결과이다. 실험에서 KVID를 SQL서버에 저장한 것보다 메인메모리 데이터베이스를 사용한 것이 메시지의 크기가 커질수록 변환 성능이 향상되었음을 알 수 있었다. XML-VMF 포맷을 이용하는 방법의 경우 파일 입출력시간으로 인하여 데이터의 크기가 커질수록 변환시간이 더욱 많이 소요됨을 알 수 있었다.

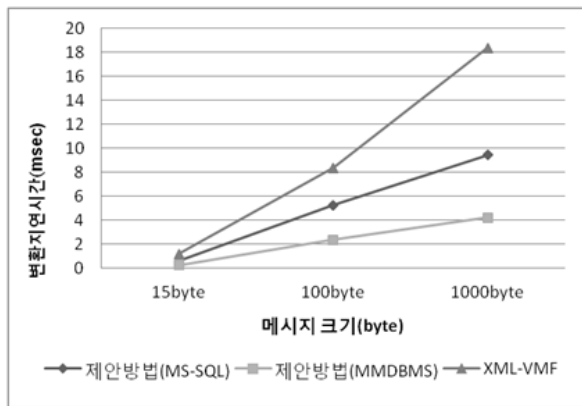


그림 10. 메시지 크기에 따른 메시지 변환성능 평가결과
Fig. 10. Evaluate result of message converting Performance according to the message sizes

IV. 결 론

네트워크 중심전의 대두와 동시에 다양한 무기체계가 등장하고 있다. 이에 따라 각 무기체계 간의 효율적인 정보교환은 전장에서 승패를 좌우하는 주요 요소이다. 전술 데이터링크는 그 핵심적인 역할을 하는데 각 무기체계 간의 효율적인 상호운용성 보장을 위해서 각기 다른 메시지 형태를 해석하여 KVMF형태로 언제, 어디서, 어떤 장비로든지 해석이 가능하여야 한다. 본 논문에서는 각 장비에 사용되는 메시지를 KVMF형태로 혹은 그 반대로 변환할 수 있는 메시지 파싱과 데이터 변환 부분에 대하여 제안하고 포병대대 사격지휘체계에 적용하여 타당성을 검증하였다. 기존 도입되어 운용중인 장비와 차기 도입될 장비들을 모두 수용하기 위하여 일관된 방법으로 각 장비의 메시지 포맷을 정의하고 문법을 생성하여 KVMF 메시지로 변환하는 방법을 적용하였으며, 차기에 도입될 장비에 대비하기 위하여 기술한 데이터 변환 모델을 적용하여 메시지 형식을 외부에 저장함으로써, KVMF 메시지로 추가적인 프로그램 수정 없이 사용할 수 있다.

참 고 문 헌

- [1] Dusseau D. and Brock C, "Network centric interoperability-using a variable message format(VMF) based data-link to improve situational awareness and close air support(CAS)," *IEEE Aerospace and Electronic System Magazine*, Vol.19(9), pp.8-13, Sept, 2004.
- [2] 국방과학연구소, *VMF급 전술데이터링크 프로토콜 연구개발제안서*, 12. 2005.
- [3] 오행록, 구흥서, *VMF 전술데이터링크 기술*, 10. 2007.
- [4] 박정운, 김영철, 김상현, 유재우, "효율적 노드 재사용을 위한 점진적 파싱 알고리즘," *한국정보과학회 논문지*, 25(2), 10월, 1998.
- [5] DOD, *MIL-STD-6017A, Variable Message Format*, July, 2006.
- [6] 임산송, 현득창, 정희경, "구조 검색을 위한 XML문서 저장 시스템," *한국정보기술전략혁신 학회지(정보학연구)*, 4, pp.89-100, 2001.
- [7] <http://redondosystems.com/vtt.html>

[8] Cicalese, Ferdinando, Gargano, Luisa, Vaccaro, Ugo, "Tunstall Parser Tree Optimum under Various Criteria," *IEEE International Symposium on Information Theory*, pp.81-85, June, 2007.

조 철 영 (Chul-young Cho) 정회원



2008년 8월 충남대학교 컴퓨터 학과 졸업
2008년 5월~현재 LIG 넥스원 소프트웨어 연구센터 연구원
<관심분야> 정보융합, 지능로봇, 패턴인식

이 준 표 (Jun-pyo Lee) 정회원



2009년 8월 한양대학교 컴퓨터 공학과 박사
2001년 3월~2003년 8월 (주)파인드테크 연구원
2003년 8월~2009년 8월 한양대학교 공학기술 연구소 연구원
2009년 8월~현재 LIG 넥스원

소프트웨어 연구센터 선임연구원
<관심분야> 영상처리, 멀티미디어 시스템, 무인지능 로봇

권 철 희 (Cheol-hee Kwon) 정회원



1998년 3월 고려대학교 제어 계측공학과 졸업
2000년 2월 고려대학교 제어 계측공학전공 석사
2000년 1월~현재 LIG 넥스원 소프트웨어연구센터 수석연구원
<관심분야> 전자공학

조 한 준 (Han-jun Cho) 정회원



1987년 2월 동국대학교 전자공학과 졸업
1990년 2월 동국대학교 전자공학과 석사
1995년 8월~현재 LIG넥스원 소프트웨어연구센터 수석연구원
<관심분야> 무인체계, 지휘통

제, 정보융합