

동적 구문처리기 소프트웨어 적용을 통한 대화력전 수행체계 연동의 유연성 향상 방안

Improving Flexibility of External Data Exchange in Count-fire Operation System
by Adapting Dynamic Parser Software

홍 원 의*

Hong, Won-Eui

ABSTRACT

The counter-fire operation system performs its mission exchanging information with other related systems such as command & control systems and military information systems. In the process of exchanging information, the counter-fire operation system uses a type of data message which contains exchange data information in the format of KMTF. The requirement of data exchange of count-fire operation will continue to evolve. But the EDX(External Data eXchange) configuration item of the current counter-fire operation system can not effectively cope with the variation of data exchange requirements due to its fixed software structure. In the paper, a solution for improving flexibility of external data exchange in counter-fire operation system is proposed.

주요기술용어(주제어) : KJCCS(Korean Joint Command Control System, 한국군합동지휘통제체계), CF(Counter-Fire, 대화력전), KMTF(Korean Message Text Format, 한국형 메시지 텍스트 포맷), EDX(External Data eXchange, 외부 자료 교환)

1. 서론

대화력전이란 우군의 탐지자산과 화력지원 수단을 통합 운용하여 적의 화력지원수단 및 간접지원체계를 무력화시키는 작전을 의미한다^[1]. 연합사로부터 대화력전 임무를 이양 받은 한국군의 대화력전 수행본부

는, 한국군 합동지휘통제체계(KJCCS : Korean Joint Command Control System)를 주 체계로 하여 지상 전술C4I체계 등 관련 체계들과의 연동 기능을 활용하는 대화력전 수행 계획을 수립하였다^[5].

대화력전 수행체계는 C4I체계, 정보체계 등과 연동하여 정보를 주고받으면서 대화력전을 수행하므로, 성공적인 작전 수행을 위하여 어느 체계보다 타 체계와의 연동이 중요시된다. 이러한 이유로 대화력전 수행체계는 연동처리 부분을 별도의 소프트웨어 형상항목으로 관리하고 있다^[6].

† 2007년 11월 7일 접수~2008년 1월 18일 게재승인

* 국방과학연구소(ADD)

주저자 이메일 : wehong@add.re.kr

연동 요구사항은 연동 대상 체계의 형상 변경이나 운용자 요구사항 추가 등의 사유로 인해 자주 변경될 수 있는데, 연동 요구사항이 변경되면 이를 반영할 수 있도록 연동 메시지가 변경되어야 하며, 연동처리 소프트웨어도 변경된 메시지를 분석, 처리할 수 있도록 변경되어야 한다.

현재의 대화력전 수행체계는 각 체계간 합의가 이루어진 한가지 형식의 연동 메시지 형식만을 고정적으로 분석, 처리하도록 구현되어 있다. 따라서 연동 요구사항이 변경되어 새롭게 개정된 연동 메시지를 처리해야 할 때마다, 연동 메시지를 처리하는 소프트웨어 컴포넌트를 재작성해야 하는 문제가 있다. 연동 요구사항은 대화력전 수행체계의 개발 기간 내 뿐만 아니라 개발이 완료되고 실제 운용 후에도 운용자 요구에 의하여 변경될 수 있는데, 이때마다 매번 소프트웨어를 수정하거나 재작성하는 것은 비용 면에서나 체계 안정성 면에서 위험 요소가 된다. 따라서 대화력전 수행체계가 새로운 연동 요구사항에 대응하도록 수정될 때 대화력전 수행체계 자체의 변경이 최소화될 수 있도록 하는 방안의 연구가 필요하다.

본 논문에서는 연동 요구사항 변경 시에도 수정을 최소화하는 소프트웨어의 구조와, 이를 이용하여 대화력전 수행체계의 연동 유연성을 향상시키는 방안을 제안한다.

2. 연동 메시지 관련연구

연동 메시지는 KMTF(Korean Message Text Format)를 형식적 포맷으로 하고 체계 간 교환 대상 자료를 내용으로 하여 구성된다. KMTF는 미군의 표준 텍스트 포맷인 MTF를 기반으로 개량한 것으로, 오랜 기간 미군의 사용을 통해 효율성과 안정성이 입증되었다고 할 수 있다. 연동 메시지와 유사하게 가상으로 구성해 본 연동 메시지 예는 그림 1과 같다.

구조적으로 INTRO, BODY, CLOSE의 세 부분으로 나뉘지며, INTRO는 메시지에 대한 헤더 정보를 담고 BODY 부분에는 주요 데이터, 명령, 승인 등의 실제 연동 자료가 담기게 되며, CLOSE는 메시지의 끝임을 표시한다. 메시지의 각 필드는 slash 기호로

구분되며 끝은 slash 기호를 두 개 붙여서 나타낸다.

```
INTRO//
EXER/1.0/M003/SYS_C4I/SYS_CF/200620323//

BODY//
S0001/123456/COMPANY/52SCH1234567890//

CLOSE//
```

[그림 1] 연동 메시지 예

예로 제시한 메시지는 C4I체계가 탐지한 표적을 대화력전 수행체계에 전달하기 위한 연동 메시지라고 가정하였다. 헤더 부분인 INTRO에는 차례대로 체계 운용모드, 버전, 메시지번호, 송신체계, 수신체계, 일시를 담고 있다. BODY 부분은 set 번호로 시작되며 이 후 필드의 데이터는 그 set이 정의한 데이터 필드에 대한 실제 값이 나타난다. 위의 예에서 S0001은 표적 정보를 나타내는 set이고 표적번호, 표적규모, 표적좌표의 데이터 필드를 가진다고 가정하였다. set 번호 다음의 세 개의 필드값은 표적번호, 표적규모, 표적좌표에 대해 매핑될 실제 데이터 값이다. 마지막에 CLOSE를 표시함으로써 하나의 연동 메시지에 대한 표현이 끝났음을 나타내게 된다.

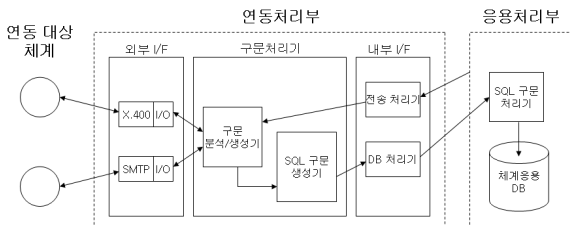
메시지의 각 필드 데이터는, 연동되는 두 체계 간의 연동 요구사항에서 교환 요소로 식별된 것이다. 만약 연동 요구사항이 변경되어 C4I체계가 대화력전 수행체계에 표적을 전달할 때 탐지시간까지 전송하여야 한다면, 표적 정보를 표현하는 set 정보가 바뀌어야 하며 이를 이용하는 메시지 또한 별도의 필드를 추가하여 표현되어야 한다.

3. 연동 처리 구조 관련연구

대화력전 수행체계의 연동 처리는 그림 2와 같이 연동처리부에 의해 수행된다. 연동처리부는 연동 대상 체계로부터 연동 자료를 수신하여, 체계의 핵심 임무를 처리하는 응용처리부가 활용할 수 있도록 가공한다.

연동처리부는 크게 내부/외부 인터페이스, 구분처리기의 부분으로 나누어진다. 외부 인터페이스는 X.400,

SMTP(Simple Mail Transfer Protocol) 등의 통신 프로토콜을 통해 연동 대상 체계로부터 연동 메시지를 받거나 연동 대상 체계로 연동 메시지를 송신한다. 구문처리는 수신한 연동 메시지를 정의된 메시지 규칙에 의해 파싱하여 그 결과를 체계 응용 DB에 적용할 수 있도록 SQL(Structured Query Language) 구문을 만들거나 체계에서 연동 대상 체계로 전송할 연동 메시지를 생성한다. 내부 인터페이스는 생성된 SQL 구문을 체계의 응용처리부 전달하거나 체계가 전송할 연동 데이터를 응용처리부로부터 받는 역할을 수행한다.



[그림 2] 대화력전 수행체계 연동처리부 구조

현재 대화력전 수행체계에서 구현된 구문처리기의 구문 분석/생성기는, 외부에서 동작 방식을 조정할 방법이 없기 때문에 각 연동 체계 간에 최종 합의된 연동 메시지의 규칙만을 처리할 수 있도록 고정하여 구현된다. SQL 구문생성기 또한 사용할 구문을 참조할 방법이 없으므로, 응용처리부 내의 체계응용 DB의 구조가 변하지 않을 것을 가정하여 고정된 SQL 구문만을 생성하도록 구현된다.

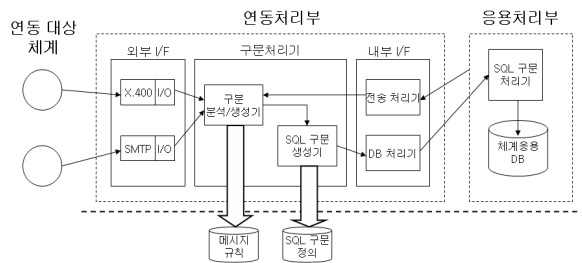
이러한 구조 하에서는, 신규 형식의 연동 메시지를 분석, 처리할 수 있도록 하려면 구문 분석/생성기의 내부 처리 로직 자체를 수정해야 하고, 응용처리부의 수정으로 인한 체계응용 DB 구조 변경 시 새로운 SQL 구문을 적용하게 하려면 SQL 구문을 정의하고 있는 SQL 구문 생성기의 내부 자료를 변경해야 하는 문제점을 가지고 있다.

4. 동적 구문처리기 설계

연동 요구사항의 변경으로 연동 메시지 형식이 바

뀌어야 하거나 응용처리부의 변경으로 체계응용 DB의 구조가 바뀌어야 될 때, 연동처리부 재작성을 최소화할 수 있어야 한다. 이러한 요구조건을 만족시키기 위해서는, 연동처리부 구문처리기 내의 구문 분석/생성기와 SQL 구문 생성기가 범용적으로 동작되게 하고, 외부에서 이들의 동작 방식을 정의할 수 있도록 하는 구조를 적용할 수 있다.

앞에서 언급된 문제점들을 보완하기 위해 이러한 구조를 적용하여 새로운 동적 구문처리기를 설계하였고 그 구조는 그림 3과 같다.



[그림 3] 동적 구문처리기 적용 연동처리부 구조

동적 구문처리기의 구문 분석/생성기는, 하나의 연동 메시지 형식만을 처리할 수 있도록 하지 않고 범용의 KMTF 메시지를 처리할 수 있도록 하며, 이를 어떻게 해석하고 처리할 것인지는 외부에서 설정과 일이나 매개변수(parameter)를 사용하여 정의한다.

동적 구문처리기의 SQL 구문 생성기도, 생성할 SQL 구문을 고정하여 내부에 갖고 있는 것이 아니라 현재의 체계응용 DB 구조에 맞게 정의한 SQL 구문을 외부에서 가져와 사용한다.

이러한 구조에서는 연동 메시지 형식의 변경이나 체계응용 DB의 변경이 발생하여도, 구문처리기를 포함한 연동처리부를 재작성하지 않고 외부에서 설정과 일이나 매개변수를 통해 동작 방식을 수정해 주는 것만으로 체계가 새로운 요구에 대응하도록 할 수 있다.

5. 메시지 처리 규칙 및 데이터 저장 방식 정의

연동처리부 외부에서 구문 분석/생성기의 동작 방식을 정의할 메시지 규칙은 연동 메시지의 필드 및

타입을 정의할 수 있어야 한다. XML 스키마가 XML 문서의 형식을 정의하듯이, 메시지 규칙은 동일한 KMTF 형식을 이용하여 연동 메시지의 형식을 정의한다. 텍스트로 정의하면 연동처리부가 시작시 그 내용을 읽어 오게 할 수 있다. 그림 4는 메시지 규칙에 대한 정의의 예이다.

그림 4의 메시지 규칙 예는 그림 1의 연동 메시지에 대하여 정의한 것이다. 메시지 정의는 버전에 대한 표시와 연동 메시지의 내용인 INTRO, BODY, CLOSE의 각 섹션에 대한 정의로 이루어진다.

```

VERSION//
1.0//

INTRO//

모드/Mode/U/4/"OPER","EXER"//
버전/Version/U/4/메시지버전//
메시지번호/MegID/U/4/메시지식별자로사용//
송신체계/SendSys/U/10/--//
수신체계/SendSys/U/10/--//
일시/SendTime/U/14/YYVYMMDDHHMMSS//

BODY//

S0001//
표적번호/targetnumber/N/6/표적번호표시(0~999999)//
표적규모/targetscale/U/5/값: "COMPANY", "PLATOON"//
표적좌표/targetcoordination/U/15/10계단좌표//

CLOSE//

```

[그림 4] 메시지 규칙 정의의 예

INTRO, BODY, CLOSE 섹션 안의 각 라인은 내부 필드에 대하여 속성을 정의한다. 즉, 한 필드에 대하여, 한글이름, 영문이름, 형식, 크기, 설명의 내용을 필드로 하여 한 라인에 명시한다. 특히, BODY에서는 각 SET을 정의하고 SET 안에서 각 SET에 대한 필드들을 정의한다. 따라서 그림 4의 정의에 의하면 INTRO의 필드는 6개이고, BODY 중 set 번호 S001에 대한 필드는 3개이다.

각 필드를 기술하는 각 라인에는 여러 정보들이 표현되게 되는데, 특히 필드의 형식과 크기를 명시해 줌으로써 실제 연동처리가 연동 메시지를 분석할 때 값에 대한 유효성 검사를 자동으로 수행하게 할 수도 있고, 설명을 명시함으로써 연동처리기 뿐 아니라 운용자도 메시지 규칙에 대한 정보를 파악할 수 있다.

그림 5는 SQL 구문 정의의 예이다.

```

S0001/
INSERT INTO Target_Info(TarNo, TarSca, TarCoord)
VALUE(':f1', ':f2', ':f3')/
UPDATE Target_Info SET TarNo=':f1',TarSca=':f2',
TarCoord=':f3'/
--

```

[그림 5] SQL 구문 정의 예시

set 번호, Create 관련 SQL 구문, Update 관련 SQL 구문, Delete 관련 SQL 구문을 slash로 구분하여 표현한다. 구문 분석/생성기에서 분석된 연동 메시지의 각 필드들이 자동으로 'f1', 'f2', 'f3'과 같은 부분에 대치되어 SQL 구문이 완성될 것이다.

이와 같은 정의를 설정 파일로 만들어 두면, 연동처리기 시작 시에 구문처리의 구문 분석/생성기와 SQL 구문 생성기가 이 설정 파일을 읽어 연동 메시지를 처리하도록 할 수 있다.

6. 동적 구문처리의 적용방안

동적 구문처리를 체계에 적용하는 방안을 하나의 시나리오를 통해 검증해 보기로 한다.

지상전술C4I체계가 대화력전 표적을 탐지했을 때 대화력전 수행체계에게 그림 1과 같이 표적번호, 표적규모, 표적좌표의 내용으로 M003의 표적보고 연동 메시지를 보내는 것으로 연동 요구사항이 식별되었다고 가정하자. 또한 대화력전 수행체계 응용처리부의 체계응용 DB에는 이 정보를 저장할 수 있도록 표적정보 테이블을 정의하여 표적번호, 표적규모, 표적좌표, 표적성질, 탐지시간 등의 표적속성을 관리하고 있었다고 가정하자.

이 때 대화력전 수행체계 운용자의 요구사항에 의해 지상전술C4I체계가 대화력전 수행체계에게 표적탐지시간을 추가로 보고하며 표적번호를 8자리로 확장하도록 하는 연동 요구사항이 새롭게 발생하면, 연동 메시지에 표적탐지시간의 정보가 추가되어야 한다.

그림 6과 같이 새로운 정보를 새로운 필드에 추가하여 대화력전 수행체계로 전송하였을 때 기존의 대화력전 수행체계 연동처리부는 그림 1과 같은 연동

메시지만 처리할 수 있도록 고정되어 구현되었으므로, 길이가 변경되거나 새로 추가된 필드를 처리하지 못할 뿐 아니라 잘못된 메시지가 수신된 것으로 처리하게 된다. 이 필드를 처리하도록 하기 위해서는 구문처리를 새롭게 작성하는 수밖에 없다.

```
INTRO//
EXER/1.0/M003/SYS_A/SYS_B/20060323//

BODY//
S0001/12345600/COMPANY/52SCH1234567890/20003221809//

CLOSE//
```

[그림 6] 변경된 연동메시지

하지만, 동적 구문처리를 적용한 대화력전 수행 체계에서는 그림 4와 같이 표현되었던 기존의 메시지 규칙을 변경하기만 하면 된다. 메시지 규칙에서 표적 번호에 대한 필드 크기를 8로 변경하고, S0001번 set에 표적탐지시간을 추가한 새로운 메시지 규칙은 그림 7과 같다.

```
VERSION//
1.0//

INTRO//

모드/Mode/U/4/"OPER","EXER"//
버전/Version/U/4/메시지버전//
메시지번호/MegID/U/4/메시지식별자로사용//
송신체계/SendSys/U/10/--
수신체계/SendSys/U/10/--
일시/SendTime/U/14/YYVYMMDDHHMMSS//

BODY//

S0001//
표적번호/targetnumber/N/8/표적번호표시(0~999999)//
표적규모/targetscale/U/5/값: "COMPANY", "PLATOON"//
표적좌표/targetcoordination/U/15/10계단좌표//
표적탐지시간/targetdetecttime/U/14/YYVYMMDDHHMMSS//

CLOSE//
```

[그림 7] 변경된 메시지 규칙

구문분석기의 구문 분석/생성기는 그림 7과 같은 메시지 규칙 설정 파일을 읽어 이후 송신하는 연동 메시지에 적용한다. 그러면 대화력전 수행체계의 연동처리부는 그림 6과 같은 새로운 메시지를 받게 되면 M003 메시지에서 표적번호가 8자리이고 표적탐지 시간 정보가 포함되어 있는 것으로 인식하고 처리하

게 된다. 이러한 방법으로 대화력전 수행체계의 연동 처리기 소프트웨어를 수정하거나 재작성하지 않고 그림 6과 같이 새로 변경된 메시지를 처리할 수 있게 된다.

만약 대화력전 수행체계가 표적규모에 대한 관리가 필요 없어져서 체계응용 DB의 표적정보 테이블에서 표적규모 속성을 삭제하고, 추가된 표적탐지시간을 체계응용 DB의 표적정보 테이블에 속성으로 추가하게 되면, 이와 관련된 정보를 생성하거나 조작할 SQL 구문도 바뀌어야 한다. 동적 구문처리를 적용한 연동 처리기에서는 SQL 구문 정의에 대한 설정 파일만 바꾸어 적용하면 연동처리를 재작성하지 않아도 된다. 이를 반영하기 위해 수정한 SQL 구문 정의는 그림 8과 같다.

표적규모가 담긴 두 번째 필드값은 DB에 반영하지 않고, 표적탐지시간이 담긴 네 번째 필드를 Target_Info 테이블 안에 반영하도록 SQL 구문을 재작성 한다.

```
S0001/
INSERT INTO Target_Info(TarNo, TarCoord, TarDetTime)
VALUE(':f1', ':f3', ':f4')/
UPDATE Target_Info SET TarNo=':f1',
TarCoord=':f3', TarDetTime=':f4'/
--/
```

[그림 8] 변경된 SQL 구문 정의

7. 결론

대화력전 수행체계는 지휘통제체계, 정보체계 등과 연동하여 대화력전을 수행하도록 개발되었다. 각 체계 간의 연동 요구사항은 체계 내외의 여러 가지 요인으로 인해 변화되기 쉬우므로, 각 체계는 이러한 변경 요인에 위험성이 적으면서도 유연한 소프트웨어 구조를 채택해야 한다.

본 논문에서 제안한 동적 구문처리를 통한 연동 유연성 향상 방안을 향후 연동을 통해 임무를 수행하는 체계에 적용하게 되면, 체계의 소프트웨어를 재작성하지 않고서도 연동 체계들이 연동 요구사항에 대응하도록 할 수 있고, 이를 통해 각 체계의 안정성과 연동 체계 간 연동 유연성을 동시에 만족시킬 수 있을 것이다.

참 고 문 헌

- [1] 화력부 대화력전과, 지구사 대화력전 전술예규, 제 3군사령부, 2006.
- [2] 국방과학연구소, 한국군합동지휘통제체계 연구, 2003.
- [3] General Dynamics, Automated Deep Operations Coordination System(ADOCS) : Integrating Information from Joint and Service C4I System, 2002.
- [4] Raytheon, AFATDS(Advanced Field Artillery Tactical Data System), 2002.
- [5] 조정재, 이용희, 한반도 지상군 대화력전 수행체계에 관한 연구(미2사단 대화력전 수행본부를 중심으로), 한미연합사령부, 軍事評論 第370號, 2004. 9.
- [6] 전기윤, 스캔라인 알고리즘을 이용한 대화력전 임기표적의 실시간 영역 결정, 韓國軍事科學技術學會誌, 2007. 9.
- [7] Jeffrey E. F. Friedl, Mastering Regular Expressions, O'Reilly, 1997.