연동통제문서 분석을 위한 객체 모델 기반의 패킷 처리 방법

심준용*, 위성혁*
*LIG넥스원(주) 소프트웨어연구센터
e-mail:junyong.shim@lignex1.com

A Method of Object Model based Packet Processing for Interface Control Document Analysis

Jun-Yong Shim*, Soung-Hyouk Wi*
*Software R&D Lab, LIG Nex1 Co., Ltd

요 약

본 논문은 개발 무기체계의 통합점검 단계에서 각 장비의 교환 데이터 분석을 용이하게 할 수 있는 연동통제문서 기반의 분석 소프트웨어 설계 구조를 기술한다. 특히, XML 형식의 객체 모델을 정의하여 수집된 패킷을 연동 메시지로 객체화하는 방법을 제안한다. 이러한 객체 모델은 메시지 교환 방법에 따라 Object와 Interaction 모델로 구분되며, 연동통제문서의 메시지에 대한 작성 규칙과 표현 형식을 정의하고 있다. 분석 소프트웨어는 객체 모델로 기술된 연동통제문서를 처리하여 실시간으로 연동 데이터를 전시하고, 체계통합점검의 결과 분석에 필요한 정보를 사용자에게 제공한다.

1. 서론

무기체계개발에 필요한 연동설계규격(Interface Control Specification)은 구성 체계 또는 장치 간 연동을 위한 연동통제문서(Interface Control Document)의 개발 기준으로 적용되며, 연동 인터페이스의 물리적 및 전기적 특성과 메시지의 송수신 규정에 대한 기준을 제공한다. 특히, 연동통제문서는 해당 장비의 연동 인터페이스 개발을 위한 메시지 목록 및 전송 구조를 기술하며, 기능적 연동과 오류탐지, 응답조건 등을 제공한다. 따라서 개발 장비의 체계통합은 연동통제문서를 기준으로 각 장비를 점검하고, 기능을 확인함으로써 완료된다.

본 논문은 개발 무기체계의 통합점검 단계에서 각 장비의 교환 데이터 분석을 용이하게 할 수 있는 연동통제문서 기반의 분석 소프트웨어의 설계 구조를 기술한다. 특히, XML[1] 형식의 객체 모델을 정의하여 수집된 패킷을 연동 메시지로 객체화하는 방법을 제안한다. 구성은 다음과 같다. 2장에서 연동통제문서를 처리하는 분석 소프트웨어의 설계 구조와 적용 패턴을 살펴보고, 3장은 객체 모델을 정의하고, 스키마 구조를 기술한다. 4장은 객체 모델을 기반으로 메시지를 변환하는 방법을 보여준 후, 5장에서 결론을 맺는다.

2. 연동통제문서 기반의 분석 소프트웨어

연동통제문서는 무기체계 소프트웨어 관리 지침에 따라 체계개발의 소프트웨어 요구사항분석 단계에서 산출되며[2], 연동개념, 방식 및 범위를 설정하고, 논리적, 물리적 연동 방법을 구체적으로 기술하며, 특히 연동 대상 간 교환되는 메시지 종류 및 프로토콜 구조 등의 세부사항을 확인할 수 있다. 한편, 무기체계 개발의 체계통합 단계에서는 장비 별 연동통제문서의 교환 메시지가 요구사항에 맞게 그 기능을 수행하는지 분석한다. 따라서 연동통제문서를 처리하는 분석 소프트웨어의 개발은 장비의 체계통합 점검을 위한 유용한 도구로 활용될 수 있다.

먼저, 논문에서 제안하는 분석 소프트웨어를 개발하기 위해서 연동 장비의 네트워크를 통해 교환되는 패킷을 수집하고, 처리할 수 있는 라이브러리가 필요하다. 일반적으로 분석 도구들은 WinPcap[3]을 기반으로 패킷 수집 기능을 구현하는데, 이를 구현한 분석 도구 중 Wireshark[4]는 산업계에서 사실 표준으로 인정될 만큼 많이 활용되고 있다. 구조는 그림 1과 같다.

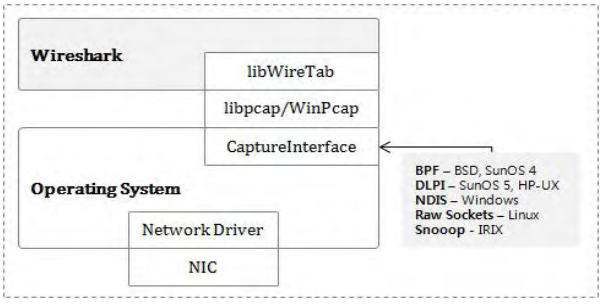

그림 1 Wireshark 구조

Wireshark는 Pcap 라이브러리로부터 수집된 패킷의 헤

더를 분석하여 통신 방법 및 프로토콜에 대한 확인을 용이하게 하지만, 사용자 데이터에 대한 분석은 어렵다

다음으로 사용자 데이터 분석을 위해서 패킷으로부터 메시지의 식별정보를 추출하여 사용자 객체로 생성할 수 있는 구조가 필요하며, 본 논문은 연동통제문서를 기반으로 그 기능을 구현할 수 있도록 공통의 인터페이스를 제안했다. 특히, 사용자 객체의 해석과 처리 방법을 분리함으로써 연동통제문서의 변경에 따른 코드 수정을 최소화하도록 했다.

분석 소프트웨어의 패킷 처리 모듈은 연동통제 메시지를 객체화하고, 파일로 추출하기 위한 공통의 인터페이스를 정의한다. 제안 구조는 그림 2와 같다.

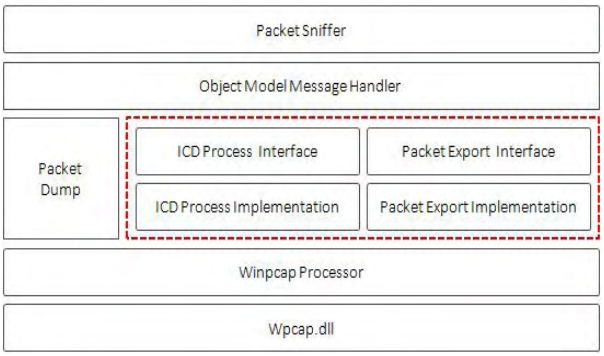

그림 2 패킷 처리 모듈 구조

제안 모듈은 Winpacap 라이브러리를 구현하여 패킷을 수집하는 Winpcap Processor, 수집된 패킷을 연동 메시지로 변환하기 위한 ICDProcessInterface, 변환 메시지를 파일 형태로 추출할 수 있도록 인터페이스를 제공하는 PacketExportInterface, 패킷을 객체화할 수 있도록 메시지 정보를 관리하는 ObjectModelMessageHandler로 구성된다. 설계 구조는 그림 3과 같다.

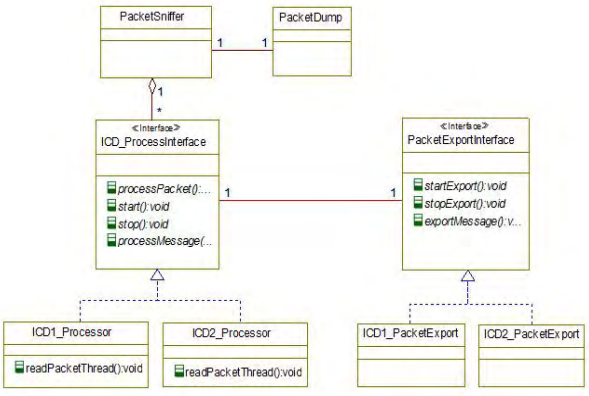

그림 3 Packet Sniffer의 클래스 다이어그램

PacketSniffer는 구성 요소를 통합하고, 연동통제 메시지를 사용자 모듈로 전달한다. 특히, ICDProcessInterface와 PacketExportInterface가 Abstract Factory 방식으로[5] 설계되어 PacketSniffer에 연동 장비가 추가되었을 때 기존 모듈의 변경 없이 추가가 용이하다. 또한, 패킷으로부터 메시지 식별자를 추출하기 위한 정보와 객체 모델의 정보를 그림 4와 같이 기술하여, 코드로부터 분리시켰다.

```
[TABLE]
COUNT = 20
ID_POS = 0
ID_LEN = 2
SIZE_POS = 2
SIZE_LEN = 2

[MSG_1]
ID = 101
NAME = Connect
TYPE = INTERACTION

[MSG_2]
ID = 102
NAME = Disconnect
TYPE = INTERACTION

[MSG_3]
ID = 103
NAME = Status
TYPE = OBJET
```

그림 4 메시지 식별 정보 테이블

만약 연동통제문서에서 요구하는 메시지의 식별번호나 이름이 변경된다면 코드 수정 없이 위의 테이블을 변경함으로써 데이터를 수집 및 분석할 수 있다.

3. 객체 모델

수집된 패킷을 객체화하기 위해 작성된 객체 모델은 메시지 교환 방법에 따라 Object와 Interaction으로 구분되며, 사용자는 연동통제문서에 정의된 메시지 교환 주기에 따라 해당 모델로 표현할 수 있다. 그림 5는 객체 모델을 표현하기 위한 스키마의 일부를 보여준다.

```
<?xml version="1.0" encoding="EUC-KR"?>
<!--Created with Liquid XML Studio Developer Edition (Trial) 8.0.11.2171 (http://www.liquid-technologies.co
- <xs:schema id="NOM" xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="lignex1
xmlns:nom="lignex1.vm.nframework.NOM">
  - <xs:element name="Nex1ObjectModel">
    - <xs:complexType>
      - <xs:sequence>
        - <xs:element name="objects">
          - <xs:complexType>
            - <xs:sequence>
              + <xs:element name="object" maxOccurs="unbounded" minOccurs="0">
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        - <xs:element name="dataTypes">
          - <xs:complexType>
            - <xs:sequence>
              + <xs:element name="basicTypes">
              + <xs:element name="enumerationTypes">
              + <xs:element name="complexTypes">
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        + <xs:element name="notes">
      </xs:sequence>
      - <xs:attribute name="version" use="required">
        + <xs:simpleType>
      </xs:attribute>
      <xs:attribute name="name" type="xs:string" use="required"/>
      <xs:attribute name="date" type="xs:date" use="required"/>
      <xs:attribute name="author" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

그림 5 객체 모델을 위한 Schema

Object와 Interaction모델은 동일한 구조를 갖지만, 교환 방법이 주기적이면 Object를, 비주기적이면 Interaction 모델을 사용한다. 가령, 장비의 플랫폼 정보나 위치 정보

는 Object를 사용하고, 이벤트 정보는 Interaction을 사용할 수 있다. 그림 6은 연동통제문서를 객체 모델로 표현한 일부를 보여준다.

그림 6 연동통제문서의 XML 형태

4. 패킷 변환 방법

수집된 패킷을 연동통제 메시지로 변환하기 위해서는 메시지 식별 정보 테이블과 연동통제문서를 작성한 객체 모델을 비교한다.

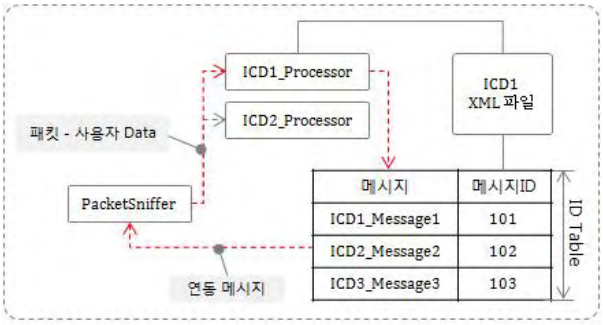

그림 7 메시지 변환 방법

그림 7과 같이 각 장비별로 작성된 ICD_Processor는 메시지 식별 정보 테이블을 통해서 필요한 객체 모델을 가져온다. 해당 메시지와 식별정보가 일치한다면, 해당 객체 모델에 사용자 데이터를 디코딩한다.

5. 결론

본 논문은 무기체계에 탑재된 소프트웨어의 통합 점검 단계에서 연동 데이터 분석을 용이하게 할 수 있는 연동통제문서 기반의 패킷 처리 방법을 제안했다. 제안 방법은 XML을 활용하여 수집된 패킷을 사용자 데이터로 객체화는 모듈을 구현했으며, 특히 메시지 식별 정보 테이블과 객체 모델을 코드로부터 분리함으로써 연동 장비의 확장을 고려했다.

참고문헌

[1] XML, http://www.w3.org/standards/xml/
[2] 방위사업청, "무기체계 소프트웨어 개발 및 관리지침" 제 2011-26호, 2011
[3] WinPcap, http://www.winpcap.org
[4] Wireshark, http://www.wireshark.org
[5] Erich Gamma, Rchard Helm, Ralph Johnson, John Vlissides, "Design Patterns: Elements of Resuable Object-Oriented Software", Addison-Wesley, 1994.