논문 2019-14-36

SRS 적합성 검증을 위한 구조화된 작성 방법 및 작성 보조 도구

(A Structured SRS Description and Its Supporting Tool for Efficient Suitability Verification of Embedded Software)

장 정 규, 이 상 훈, 양 회 석*
(Jeonggyu Jang, Sanghoon Lee, Hoeseok Yang)

Abstract : Today's embedded software (SW) developments are mostly preceded by composing Software Requirement Specification (SRS). In particular, in the domain of weapon systems, it is essential to have a systematic method for the verification of the SW functionality. To be more specific, it is crucial to check if the SW functionality is implemented as described in SRS, so-called SW suitability verification. Unfortunately, existing static or dynamic SW testing methods are not sufficient to evaluate suitability with SRS since those testings only verify the robustness of the SW codes. In this paper, we propose an automatic embedded SW suitability verification framework which is based on a structured SRS. The major challenge in the automation of this verification framework is how to get rid of ambiguities in SRS. In order to overcome this challenge, we propose a structured SRS description framework and the supporting toolchain for that. We show how the proposed framework is applied to an actual SRS of a weapon system.

Keywords : Software requirement specification, Software testing, SW suitability verification, Test case generation, Simplified Technical Korean (STK)

Ⅰ. 서 론

한국군 무기체계 SW (Software) 개발은 무기체계 SW가 만족해야할 요구사항을 명세하는 SRS (Software Requirement Specification) 작성이 선행되어야 한다. SRS는 SW 개발자가 정확하게 요구사항을 이해할 수 있도록 명확하게 작성되어야 한다. 하지만, 기존 SRS의 작성은 방위사업청에서 발간한 소프트웨어요구사항명세서 작성방법 [1]에 따라 표준 양식을 기반으로 작성되지만 문장의 명확성을 위한 문장단위의 작성 지침은 별도로 존재하지 않고 자연어 (한국어)를 그대로 따른다. 따라서 자연어로 작성된 SRS의 요구사항 문장이 갖는 모호성은 SW 개발자가 해석할 때, 잘못된 해석을 초래할 수 있으며 요구사항을 만족하지 못하는 무기체계 SW 개발로 이어질 수 있다.

한편, 개발된 무기체계 SW의 검증을 위해서 일반적으로 소스코드의 강건성을 검사하는 정적/동적 검증기법들이 사용된다 [2, 3]. 하지만, 소스코드의 강건성을 검사하는 수단일 뿐, SW가 SRS의 요구사항들을 얼마나 만족하는지에 대해서는 검증하지 못한다. 따라서, 개발된 무기체계 SW의 SRS 적합성 검증이 필요하다. SRS 적합성 검증은 SRS를 기반으로 생성한 테스트케이스를 사용한다. 하지만, 테스트 케이스 생성 또한 SRS를 해석하여 진행하기 때문에 앞서 설명한 자연어로 작성된 SRS의 모호성은 테스트케이스의 신뢰도에도 영향을 준다.

자연어로 작성된 SRS가 갖는 이러한 문제점들을 해결하기 위해서는 좀 더 명확한 SRS가 필요하다. 유사한 문제는 영어에서도 발생한다. 유사한 문제해결을 위해서 자연어로 작성되는 기술문서의 단어와 문장 구조를 제한하여 모호성을 줄이는 여러 가지 형태의 CNL (Controlled Natural Language)

*Corresponding Author (hyang@ajou.ac.kr)
Received: July 12, 2019, Revised: July 30, 2019, Accepted: Aug. 23, 2019.
J. Jang, S. Lee, H. Yang: Ajou University.
※ 본 논문은 국방과학연구소의 지원(계약번호 : UD170016DD)을 받아 수행하였음.

ⓒ IEMEK J. Embed. Sys. Appl. 2019 Dec. 14(6) 329-338
ISSN : 1975-5066
http://dx.doi.org/10.14372/IEMEK.2019.14.6.329

이 제안되었다. 유럽과 미국의 항공 및 국방 산업에서는 CNL을 활용한 STE (Simplified Technical English)를 사용하여 정비, 유지보수 관련 기술 문서에 대한 작성 규칙을 다루고 있다. 하지만, STE는 영문법을 기반으로 개발되었고 그 목표 또한 SW 요구사항 문서가 아니기 때문에 한국군 무기체계 SRS에 바로 적용하기는 어렵다.

본 논문에서는 SRS 적합성 검증을 위한 구조화된 SRS 작성 방법 및 작성 보조 도구를 제안하고 실제 기존 무기체계 SRS에 적용한 결과를 기반으로 그 가능성을 보인다. SRS 작성 방법은 기존 작성된 무기체계 SRS를 분석하여 SRS 작성에 사용되는 주요 문장들과 구조를 분석하고 이를 최대한 수용할 수 있도록 구조화한다. 구조화된 SRS는 문장 단위 구조화와 SRS 구성에 대한 구조화, 총 두 개의 구조화 단계로 나뉜다. 문장단위 구조화를 위해서는 CNL의 일종인 STE를 기반으로 한국어 문법과 무기체계 SRS에 특화된 STK (Simplified Technical Korean)를 제안하고 SRS 구성에 대한 구조화를 위해서는 무기체계 SRS 작성 지침 표준을 활용한다. 이를 통해 SRS의 명확성이 향상되고 개발된 무기체계 SW의 SRS 적합성과 생성된 테스트케이스의 신뢰도가 향상된다. 또한, 구조화된 SRS는 프로그램을 통한 자동화 가능성을 높이기 때문에 시스템적인 SRS 분석이 가능하다. 이는 테스트케이스 생성 자동화로 이어질 수 있다. 마지막으로 본 논문에서는 제안한 SRS 작성 방법을 기반으로 요구사항을 작성할 때, 작성 방법에 대한 이해의 필요성을 고려하여 이를 돕기 위한 작성 보조 도구를 제안한다. SRS 작성 시 작성 방법에 대한 오류 검사를 통해 SRS의 완성도를 높일 수 있다.

Ⅱ. 관련연구

1. CNL (Controlled Natural Language)

요구사항과 같은 기술문서를 자연어로 작성하는 경우 발생하는 해석의 모호성을 줄이기 위하여 기술문서를 작성할 때 자연어 문법이나 사용되는 단어를 제한하는 방법인 CNL이 존재한다. CNL은 이미 영어, 프랑스어, 독일어, 일본어 등으로 정의되어 있으며 사람과 사람의 정보 전달 뿐 아니라 기계와의 소통에도 용이하다. 구조화된 정보는 프로그램을 통한 분석이 가능하므로 자동화의 장점이 있다. 이미 해외에서는 학술연구, 국방, 항공우주, 비즈니스 등 다양한 분야에서 사용되고 있다 [4].

일반적으로 CNL은 문서 작성에 사용되는 자연어 문법 부분과 기술문서에서 사용하는 단어의 해석과 사용방법을 제한하는 부분으로 구성된다. 단어에 대한 부분은 사용되는 분야에 따라 정의하여 사용하면 되지만 문법에 대한 부분은 사용되는 자연어에 따라 크게 달라질 수 있다. 특히, 한국어 문법은 다른 언어와 문법적인 차이가 크기 때문에 기존 CNL을 한국어에서 사용하는 것은 바람직하지 않다.

2. STE (Simplified Technical English)

STE는 영어로 작성하는 기술문서에서 사용할 수 있는 CNL이다. 독일 항공산업 회사인 Fokker에서 최초로 개발되었으며 유럽의 항공, 국방 분야인 ASD (Aerospace and Defense) [5] 산업에서도 사용되고 있다. 최근에는 미국과 일본 등 다양한 나라와 전자, 자동차 산업 등 다양한 분야에서도 많이 사용되고 있다. STE는 사용되는 분야에 특성에 따라 STE에서 제공하는 사전에 등록되지 않은 단어들을 추가할 수 있고 간단한 문법을 사용한다는 특징이 있다. STE는 일반적인 CNL의 구성을 따라 작성 규칙 (writing rule)과 사전 (dictionary)으로 작성되어있다. 작성 규칙은 총 55개가 있으며 대부분 품사를 기반으로 문장 작성에 대한 규칙을 다루고 있다. 사전의 경우 STE에서 허용하는 명사와 동사의 단어들을 나열하고 각 단어들의 사용방법과 해석방향을 제한한다. STE는 기술문서 작성에 필요한 단어 수를 줄이고 간결한 문장을 작성하도록 하여 문서의 길이 또한 줄어들 수 있으며 해석의 모호성을 줄일 수 있다는 연구 결과 [6]가 존재한다.

Ⅲ. 구조화된 SRS 작성 방법

기본적으로 자연어가 갖는 자율성으로 인하여 해석의 모호성이 발생하는 문제는 SRS를 기반으로 개발과 검증이 진행되는 무기체계 SW 개발에서는 큰 문제가 될 수 있다. 또한, 일반적으로 SRS를 기반으로 SW 검증이 진행되지 않고 개발된 코드의 강건성 위주의 검증이 진행되는 상황에서 이러한 모호성 문제는 더욱 큰 문제를 발생시킨다. 예를 들어 SW 개발자가 SRS를 잘못이해하고 개발이 진행되었을 경우 최종 SW 검증 단계에서, 개발된 코드의 강건성 위주로 검사를 진행한다면 코드의 완성도에 따라서 검증이 통과되더라도 SRS와는 상이한 결과물이 도출될 수 있다. 따라서 SRS의 모호성을 줄이기 위한 대책은 필수적이다. 본 논문에서는 이

표 1. STK Rule (체제 규칙)
Table 1. STK Rule (Structure Rule)

규칙 1.1	요구사항은 소프트웨어요구사항명세서 작성방법을 따른다.
규칙 1.2	유효한 요구사항은 식별자가 포함된 표안에 기술한다.
규칙 1.3	단위 SRS의 행동 명세는 번호 체제에 따라 작성한다.

표 2. STK Rule (작성 규칙)
Table 2. STK Rule (Writing Rule)

규칙 2.1	서술어의 시제는 현재형만 사용한다.
규칙 2.2	서술어는 “~한다”를 사용한다.
규칙 2.3	조건문은 “~경우”로 작성한다.
...	...

를 해결하기 위해서 구조화된 SRS 작성 방법을 제안한다. 구조화된 SRS는 문장단위 구조화와 SRS 구성에 대한 구조화로 나뉘고 본 절에서 두 내용을 차례로 다룬다. 또한 구조화된 SRS를 통해 모호성을 줄임과 동시에 SRS 기반의 테스트 케이스를 생성할 때 객관적인 테스트 케이스 생성을 도울 수 있으므로 SRS 기반 검증을 도울 수 있다.

1. STK

STK는 SRS를 작성하기 위한 기본 명세지침인 STK Rule과 SRS의 요구사항 문장들에 대한 문장단위 작성 규칙인 STK Template으로 구성되었다. STK는 기본적으로 STE를 기반으로 정립되었으며 기존 무기체계 SW 개발에서 사용된 SRS를 분석하여 기존 틀을 크게 벗어나지 않는 것을 목표로 한다. 따라서, 기존 SRS에 익숙한 관계자들에게 구조화된 새로운 SRS의 수용이 어렵지 않도록 한다.

STK Rule은 요구사항 작성의 전반적인 체제를 다루는 3개의 체재 규칙과 구체적인 작성에 대한 제한사항들을 다루는 16개의 작성규칙으로 구성되어있다. STE는 국제적으로 인증된 표준으로 사용되기 때문에 STK 또한 STE를 기반으로 작성한다. 기본적으로 STE의 내용은 영문법에 맞추어져있으므로 한국어 문법에 적용이 어려운 내용들이 다수 존재한다. 따라서, STE의 규칙들 중에서 한국어 문법에 존재하지 않는 내용들을 삭제한 뒤, 남은 규칙들에 대해서는 기존 SRS에 대응해본 뒤 기존 작성된 SRS에 내재된 규칙에 부합하는지 파악한 뒤 STK로 추가한다. 추가적으로 STE에는 존재하지 않지만 기존 SRS에서 일반적으로 나타나는 규칙과 한국어 문법의 특수성을 고려한 규칙들이 존재한다.

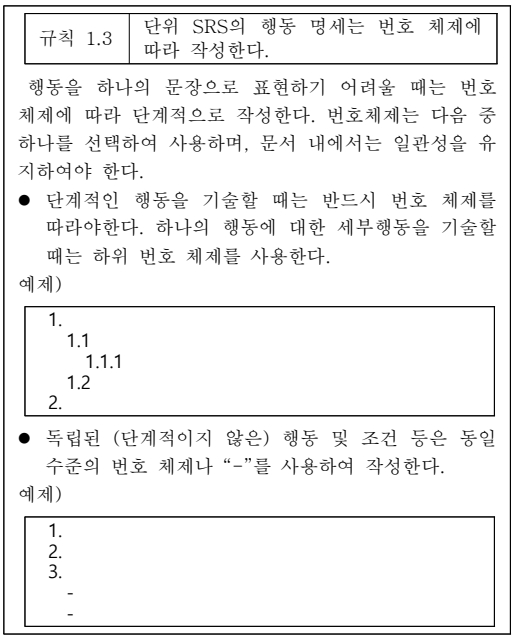

그림 1. STK – 체제 규칙 예시 (체제 규칙 1.3)
Fig. 1 STK – Structure Rule Example (Rule 1.3)

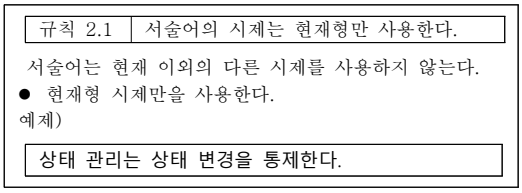

그림 2. STK – 작성 규칙 예시 (작성 규칙 2.1)
Fig. 2.STK – Writing Rule Example (Rule 2.1)

무기체계 SRS가 갖는 전문성과 새로운 SRS를 작성할 때 이전 SRS를 참고할 수밖에 없으므로 고착화되어 나타나는 규칙들이 이러한 예이다.

표 1과 2는 STK Rule의 예시이다. 체제 규칙은 요구사항에 작성될 문장들에 대한 지침이 아니라 전체 요구사항 작성에 있어 전반적으로 적용되는 체제에 대한 규칙을 다룬다. 작성 규칙의 경우 한국어 문법을 기반으로 문장을 구성하는 절, 구, 형태소 등의 단위로 규칙을 명세한다. 해석의 모호성을 배제하기 위한 직접적인 접근방식이며, 여러 가지 표현방법에 대해서는 기존 작성된 SRS에서 사용되는 문장들이 최대한 허용되도록 규칙을 정립하였다.

그림 1은 체제 규칙 1.3의 내용이다. 체제 규칙 1.3은 단위 SRS의 행동 명세를 위한 번호 체제와 관련된 규칙이다. STK Rule의 체제 규칙을 작성하

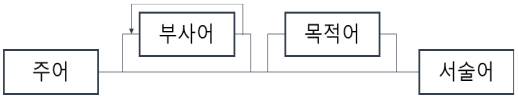

그림 3. STK Template 일반문장1
Fig. 3 STK Template Normal Sentence 1

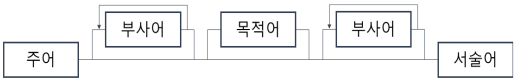

그림 4. STK Template 일반문장2
Fig. 4 STK Template Normal Sentence 2

는 방법은 규칙 번호와 제목으로 구분되도록 가장 위에 작성하고 해당 규칙에 대한 설명을 뒤이어 작성한다. 그 다음 해당 규칙에 대한 세부적인 사항들이 나열된다. 세부사항들의 경우 설명 문장과 예시를 통해 나열되며 실제 작성된 STK Rule에는 기존 무기체계 SRS의 내용을 토대로 예시가 작성되어있다. 이를 통해 기존 SRS와 관련된 개발자와 검사자 모두 쉽게 파악할 수 있도록 하였다. 그림 2는 작성 규칙 2.1의 내용이며 구성은 체제 규칙과 같다.

STK Template은 SRS를 작성할 때, 요구사항 문장에 대한 틀을 제공하는 것으로 STK Rule에서의 작성 규칙과는 다르다. STK Rule의 작성 규칙이 요구사항 문장의 제한적인 규칙만을 명세하고 규칙이 위배되지 않는 선에서 작성이 자유로운 반면, STK Template은 한국어 문법의 문장 구성단위를 기반으로 고정된 예시 문장구조를 제안하고 SRS 작성자는 손쉽게 예시 문장들을 선택하여 문장을 작성하면 STK를 만족하는 문장을 작성하기 쉽다. 자연어로 문장을 작성할 때 규칙을 별도로 파악하는 노력을 크게 줄여줄 수 있는 수단이다.

그림 3은 STK Template의 일반문장 1에 대한 그림이다. STK Template은 그림처럼 주어, 목적어, 부사어, 서술어 등의 기본적인 문장 성분들로 구성되어 있으며 각 문장 성분에 대한 세부 설명 및 규칙 또한 별도로 제공된다. 예를 들어 주어에 대한 설명은 명사 (구/절) + [이/가, 은/는]으로 나타내고 한국어에서 허용되는 다른 주어 표현은 모호성을 줄이기 위해 무기체계 SRS 작성에서 제외된다. 기본적으로 한국어 문법에 근거하여 작성되었기 때문에 별도의 큰 노력 없이 문장 성분에 대한 이해가 가능한 수준이며, 설명 또한 명확하고 쉽게 예시문장들과 함께 제공되기 때문에 이해가 쉽다. STK Template은 일반문장 2개, 특수문장 2개, 조건문장 1개로 구성되어있으며 조건문장은 일반문장과 특수문장을 활용한다.

표 3. 연결 정보 표 예시
Table 3. Mapping Information Table Example

식별자	표 번호	표 양식 번호
SRS-1	2	1
SRS-2	5	2

그림 3과 그림 4를 함께 보면 그림 4의 일반문장2가 그림 3의 일반문장1을 포함한다고 볼 수 있다. 하지만, STK Template은 문장의 구조와 더불어 해석에 대한 제한도 함께 담겨있다. 예를 들어, 일반문장1의 부사어는 서술어를 수식하는 용도로만 사용된다. 일반문장2의 부사어는 순서대로 뒤이어오는 목적어와 서술어를 수식한다. 따라서 일반문장2로는 일반문장1의 수식구조를 표현할 수 없다. 모든 STK Template은 해석의 방법이 정해져있고 이를 통해 프로그램을 통한 자연어 의미 분석의 어려움을 극복할 수 있다.

2. SRS 구성

기존 SRS의 작성은 방위사업청에서 발간한 소프트웨어요구사항명세서 작성방법 [1]에 따라 작성되고 있다. 해당 양식에서는 SRS의 전체 개요와 요구사항 작성을 위한 표 양식 등을 제공한다. 주요 구성으로는 요구사항 식별자로 구분되는 요구사항표와 여러 요구사항에서 사용되는 별도의 정보를 담은 표가 있다. 하지만, 방위사업청 양식 [1]에 따라 작성된 기존 SRS는 요구사항표와 별도의 정보표와의 관계를 명확히 명세하지 않는다. 따라서, 요구사항에 대한 개발과 검증단계에서 필요한 정보를 명확하게 파악하기 어렵다. 따라서 본 논문에서 제안하는 구조화된 SRS는 요구사항 기반의 개발과 검증에서 더욱 명확한 정보를 파악할 수 있는 SRS 구성의 개선방향을 제안한다.

요구사항표의 경우 방위사업청의 양식 [1]을 그대로 사용한다. 하지만, 기존 별도의 정보 표의 경우 내용에 따라 각기 다른 양식을 자유롭게 사용하고 있기 때문에 해석의 모호성이 존재하고 구조화하가 어렵다. 따라서 무기체계 SRS에서 주로 사용되었던 표 양식을 기반으로 표준 표 양식을 제안한다. 해당 표 양식의 경우 필요에 따라 더욱 추가될 수 있으며 현재 3개의 표 양식만으로 기존 SRS의 별도의 정보 표의 내용을 표현할 수 있다. 또한, 별도의 정보와 요구사항간의 관계를 명확히 나타내기 위한 연결 정보 표를 제안한다. 표 3은 연결 정보 표의 예시이다. 연결 정보 표는 요구사항의 식별자로 구분되고 그에 따라 연관된 별도의 정보 표의 번호를 제공한다. 또한, 해당 표의 표 양식을 함께

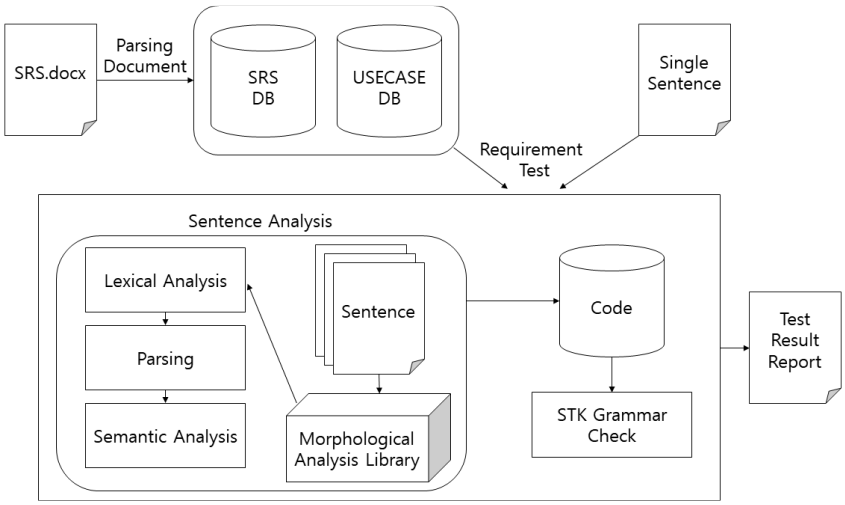

그림 5. 작성 보조 도구 개요도
Fig. 5 Description Supporting Tool Overview

제공한다. 표 양식은 STK Template과 유사하게 표를 해석하는 방법이 정해져있으므로 모호성을 줄이고 프로그램을 통한 분석에 용이하다.

Ⅳ. 작성 보조 도구

본 문단에서는 SRS가 정확히 STK에 근거하여 작성되었는지 검증하는 검사 도구를 제안한다. SRS 작성자는 이 작성 보조 도구를 통해 각 문장이 STK Template에 부합하는지 구문 분석 결과를 출력 받은 후 올바른 작성 지침에 따라 해당 문장을 교정하여 존재 가능한 모호성 또는 부정확성을 배제하고 형식을 타 문장과 통일시켜 전반적인 SRS 품질을 향상시킬 수 있다. 이에 더하여, 도구는 특정 형식을 따르는 구조화된 SRS 문서에 대한 일괄 검사 또는 단일 문장에 대한 제한적 검사 모두에 대한 두 가지 입력 인터페이스를 지원하여 이러한 프로세스를 원활히 수행하도록 돕는다.

작성 보조 도구의 전반적인 동작 다이어그램은 그림 5와 같다. 먼저, STK 검사 과정에 앞서 한국어의 일반적인 통사 구조에 기반한 문장 구조 분석 과정을 수행하여 입력 문장을 구조화한다. 이 과정을 통하여 주어, 서술어 따위의 문장 성분이나 문장과 문장 간의 포함 관계 등을 분석할 수 있다. 이후, 구문 분석 결과를 STK Template과 일대일 대응하여 부합 또는 위배 사항을 판별한다. 이러한 하향식 알고리즘은 문장 분석 과정과 STK Template 대조 과정을 분리함으로써 STK Template 개정에 의해 부수적으로 발생할 수 있는 유지보수 비용을 효과적으로 절약하는 데 장점을 지닌다.

1. 문장 분석

자연어 문법의 자유도를 제한함으로써 문장을 구성하는 각 성분에 대한 모호성을 배제하고 규칙화할 수 있다. 본 작성 보조 도구의 목표는 단순히 해당 문장이 STK Template에 부합하는지를 검사하는 데 있으므로, 분석 과정에서 해당 문장을 구성하는 단어들의 추상적 의미는 고려하지 않는다. 따라서 문장 분석 과정은 단계적 절차로, 크게 '어휘 분석', '구문 분석', '의미 분석'의 순서를 따른다.

어휘 분석 (lexical analysis)은 의미 있는 문법 단위인 토큰 (token)을 분리하는 과정으로, 한국어에서는 형태소에 해당한다. 형태소는 '뜻을 가진 가장 작은 말의 단위'로, "컴퓨터는 운용자의 요청에 따라 데이터를 처리한다."라는 예시 문장에서 형태소 목록은 그림 6과 같다. 따라서 토큰을 선별하기 위해 형태소 분석 과정이 우선되어야 하며, 본 연구에서는 KOMORAN [7] 라이브러리를 활용하였다.

토큰을 분리하는 목적은 구문 분석 단계에서 구문 트리를 형성하기 위함이므로, 문장 성분을 구분하는 데 필요하지 않은, 세부적으로 나뉘어진 품사 종류는 정리되어야 한다. 또한 체언의 경우 여러 형태소가 모여 한 단어를 형성하는 경우가 많고, 용언의 경우 용법에 따른 어미의 변화가 많으므로 별개

그림 6. KOMORAN을 이용한 형태소 분석 결과
Fig. 6 Morpheme Analysis using KOMORAN

표 4. 체언, 용언에 대한 토큰화 절차
Table 4. Tokenization Procedure for Noun and Verb

품사	절차
체언	1. 괄호 안의 문장을 체언으로 처리 2. 품사 간결화 3. 체언에 붙은 접두사, 접미사 결합 4. 체언에 연결된 문장 부호 처리 5. 띄어 쓰이지 않은 체언 형태소들을 결합 6. '경우', '때' 따위의 시간 부사 처리
용언	1. 품사 간결화 2. 선어말 어미 결합 3. 보조 용언 결합 4. 종결 어미 처리

의 항목으로 분리하여 다루는 절차가 요구되는데, 그림 6에서의 {'운용', '자'} 또는 {'하', 'ㄴ다'}가 해당되며 이에 대한 처리 과정을 표 4에 간략히 나열하였다. 또한 표 5는 위 예시 문장에 대하여 어휘 분석 단계를 수행한 결과를 나타낸 것이다.

표 5. 토큰 리스트
Table 5. Token List

토큰	구분	비고
컴퓨터	체언	
는	조사 (주격)	보조사
운용자	체언	
의	조사 (관형격)	
요청	체언	
에	조사 (부사격)	
따르	용언	부사절
아	어미 (부사격)	
데이터	명사	
를	조사 (목적격)	
처리	명사	어근
한다	용언	접사, 종결 문장

구문 분석 (syntactic analysis)은 토큰 간의 관계가 문법적으로 올바른지 검사하고 구문 트리를 형성하는 과정이다. 교착어의 특성을 지닌 한국어에서 단어의 뒤에 붙는 조사는 문법적으로 중요한 역할을 수행하는데, 그 중 격조사의 경우 체언의 뒤에 붙어 일정한 자격을 부여한다. 예로 들어, 표 5의 '컴퓨터', '운용자', '요청', '데이터'는 뒤의 조사에 의해 각각 주어, 관형어, 부사어, 목적어 등의 문장 성분이 된다. 또한 한국어 문장은 서술어로 끝맺으므로, 서술어를 구성하는 어미의 형태를 통해 그 문장의 성분을 판별할 수 있는데, 표 5에서 '따르'는 '아'에 의해 부사절을 형성하는 것이 그러하다. 이를 통해 문장과 문장 간의 포함 관계를 적절히 구별할 수 있다. 조사는 체언의 문장 성분을 분별하고, 용언 어미는 문장의 문장 성분을 분별하며, 이러한 상하 관계는 구문 트리를 형성하는 기준이 된다.

구문 분석 단계는 스택 구조를 할당하고 문장의 처음부터 토큰을 스택에 추가하기 시작하여, 문장 성분을 부여하는 조사나 어미가 들어오면 앞의 성분을 제거하고 비교하여 결합한 뒤 다시 추가하는 형태로 진행된다. 또한 서술어가 들어오면 새로운 스택을 생성하고 현재까지 형성된 스택을 그 안에 담는다. 즉, 스택은 하나의 문장이 되는 것이며, 안긴 문장이 존재하면 스택이 중첩된다. 이러한 방식으로 문장의 끝까지 탐색을 마쳤을 때, 스택의 노드 한 개를 구문 트리의 노드 한 개에 대응하여 구문 트리를 생성할 수 있다. 표 5의 결과를 구문 분석하여 형성한 스택 구조는 그림 7과 같으며, 이를 토대로 생성한 구문 트리는 그림 8과 같다.

의미 분석 (sematic analysis)은 구문 분석 단계에서 얻은 구문 트리에 대하여 의미를 부여하는 마지막 과정이다. 한국어는 어순이 자유로운 언어이므

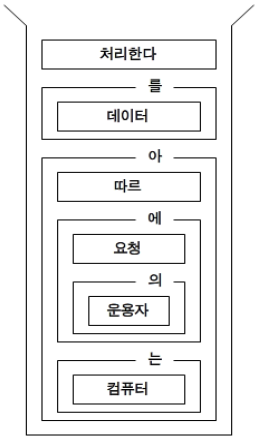

그림 7. 표 5의 토큰 리스트로부터 생성된 스택 구조
Fig. 7 The Stack Structure Generated from The Token List of Table 5

로 문장 구조 판별에 있어 의미 해석이 불가피하다. 이는 문장이 길어지고, 복합적으로 구성될수록 중요하다. 의미 분석 단계에서는 최소한의 문장 성분을 활용하여 문장과 문장의 경계를 구분하고, 문장 간의 대등 관계를 판별하기 위한 기초적인 기법을 적용하여, 구문 분석 단계에서 생성된 구문 트리를 적절히 교정하는 데 목적을 둔다.

가령, 위의 예시 문장 “컴퓨터는 운용자의 요청에 따라 데이터를 처리한다.”에서 주어 ‘컴퓨터는’에 대응되는 서술어는 ‘따르다’가 아니라 ‘처리하다’이어야 한다. 따라서 그림 8의 구문 트리는 잘못 구성된 것임을 알 수 있으며 그림 9와 같이 수정되어야 한다. 또 다른 문장 “컴퓨터는 운용자가 요청한 데이터를 처리한다.”를 보자. ‘컴퓨터는’, ‘운용자가’는 모두 주어 성분이나, 후자는 {‘운용자가’, ‘요청하다’}에, 전자는 의미적으로 {‘컴퓨터는’, ‘운용자가 요청한 데이터를’, ‘처리하다’}의 문장에 대응되는 문장 성분이다. 따라서 두 주어 성분을 구분해서 처리하기 위한 적절한 기법이 필요하다.

보조사 ‘은/는’은 주어, 목적어, 보어 모두에 대해서 쓰일 수 있으므로 의미 해석 없이 문법적 관계를 파악할 수 없으며, 강조 등의 역할만 수행할 뿐이다. 그러나 주어를 형성하는 데 주로 활용되며, 한 문장 안에 다른 문장이 안겨 있는 경우 안은 문장에는 ‘은/는’이, 안긴 문장에는 ‘이/가’가 오는 것이 일반적이라는 점에 주목할 때, 위 문제는 구문 트리에서 보조사가 붙은 주어들을 이동시키는 방식으로 해결 가능하다.

표 6. 생성 코드
Table 6. Generation Code

문장 0	문장 성분	안긴 문장
컴퓨터는	주어	
운용자의 요청에 따라	부사어	문장 1
데이터를	목적어	
처리하다	서술어	

문장 1	문장 성분	안긴 문장
운용자의 요청에	부사어	
따르다	서술어	

또한, 문장 “STK Rule 검사로써 모호성이 포함된 부적절한 부분을 배제한다.”에서 의미적으로 ‘STK Rule 검사로써’라는 부사어는 서술어 ‘배제하다’와 연결되는 것이 자연스럽다. 그러나 구문 분석 단계에서는 문장의 첫 부분부터 읽어 나가면서 서술어가 들어오면 이전에 축적된 스택을 문장으로 처리하는 알고리즘 방식을 따르므로 서술어 ‘포함되다’에 대응되지 않도록 적절한 처리가 요구된다. 문장 “운용자의 요청에 따라 갱신해야 한다.”에서 ‘따라’와 ‘갱신하다’의 경우와 같이 부사어 뒤에 바로 서술어가 오는 경우, 연결 관계를 파악하기 쉬워 문장 성분을 자연스럽게 분별할 수 있으나, 이전과 같이 부사어와 서술어가 떨어져 있는 경우, 연결 관계를 분별하기 어려운 문제가 존재하는데, 이는 부사어가 길어지면 앞에 두는 한국어의 경향에 의한 것이다. 본 연구에서는 후자에 관한 문제점을 최소화하기 위해, 이러한 부사어를 구문 트리에서 최상위 문장의 구성 성분이 되도록 위치를 조정한다.

이 외에도 문장과 문장이 대등한 관계로 연결된 형태인 경우, 접속사를 기준으로 왼쪽과 오른쪽에 있는 안긴 문장을 포함한 여러 문장들을 대조하고, 복합적으로 구성되어 있는 정도를 비교하여, 비슷한 수준으로 맞추는 방식으로 모호성을 줄일 수 있다.

위 논의의 대상은 주로 복합 문장이며, 단어의 조합을 고려하여 문맥을 파악해야 하는 자연어의 특성에 의한 것이므로, 모호성을 어느 정도 소거할 수는 있으나 무결성을 보장할 수는 없다. 따라서 SRS 작성 시 STK에 따라서 최대한 복합문장의 사용은 피하도록 권고한다.

2. STK 검사

의미 분석 단계까지 완료하면 문장 구조 분석을 마치고 STK 검사를 위한 코드를 생성하게 된다. 코드는 각 문장 단위로 생성되며, 처음의 예시 문장 “컴퓨터는 운용자의 요청에 따라 데이터를 처리한

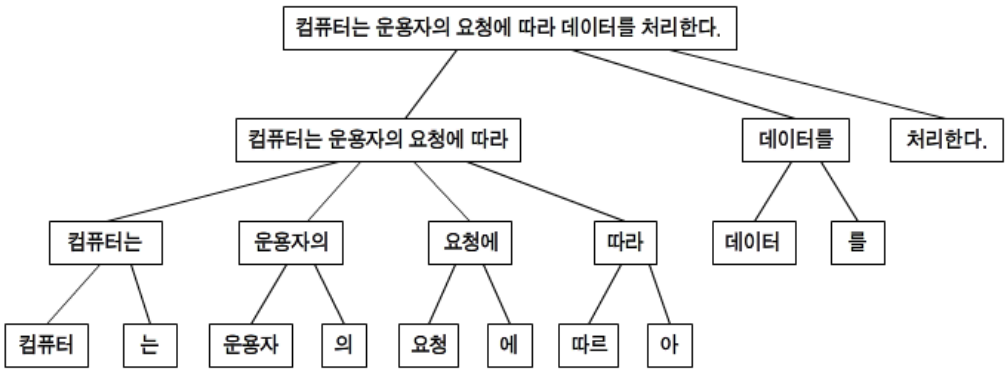

그림 8. 그림 7의 스택 구조에 대응되는 구문 트리
Fig. 8 The Syntax Tree Corresponding to The Stack Structure in Figure 6

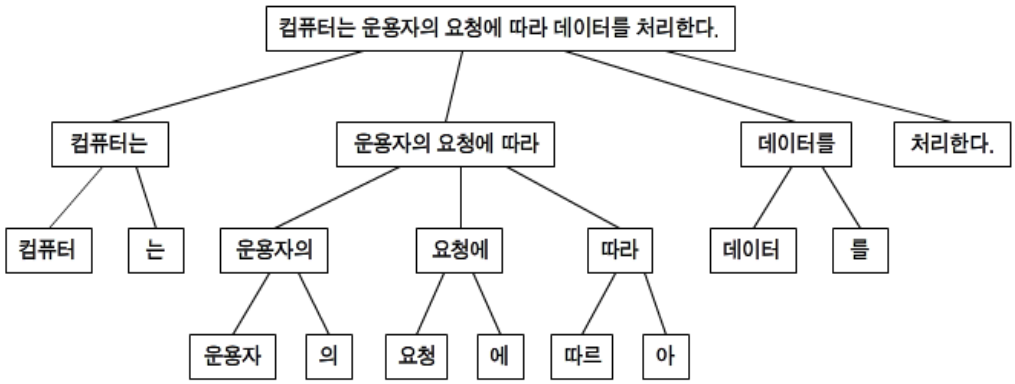

그림 9. 수정된 구문 트리
Fig. 9 Modified Syntax Tree

다."에 대한 코드는 표 6과 같다.

생성 코드는 문장과 각 문장 성분, 그리고 그 성분이 포함하는 안긴 문장에 대한 포인터로 이루어져 있으며, 이 코드를 정규 표현식을 활용하여 STK Rule과 대조한 후, 부합 또는 위배 사항을 확인하는 과정을 따른다. 예를 들어, 주어, 서술어, 목적어, 부사어를 각각 S, V, O, A라 할 때, STK template 일반 문장 1 (그림 3 참조)의 정규표현식은 SA*OV가 되고, 문장 0은 SAOV, 문장 1은 AV가 된다. 이 때 문장 1은 안긴 문장임을 고려하여 주어, 목적어의 유무를 예외로 둘 때, 문장 0, 1은 모두 STK template에 부합하다는 결과를 도출할 수 있다.

Ⅴ. 작성 보조 도구

1. STK

제안한 STK는 실제 국방과학연구소에서 작성되었거나 현재 작성단계인 무기체계 SRS 두 편에 적용하였다. 작성된 원본에서 사용된 문장을 주어, 서술어, 목적어, 부사어로 분류한 경우 총 12개의 문장 패턴을 찾을 수 있었다. 제안된 STK를 적용한 결과 5개의 STK template으로 모든 문장을 재서술

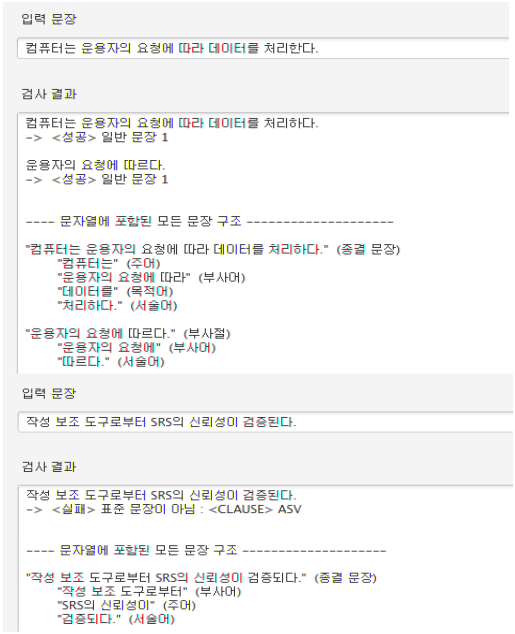

그림 10. 문장 검사 결과
Fig. 10 Sentence Check Result

할 수 있었으며, STK template을 통해 의미분석이 통제되기 때문에 명확한 정보전달이 가능했다.

또한, STK로 작성된 구조화된 SRS로부터 테스트 케이스 생성에 필요한 정보를 자동화하여 추출하는 실험을 진행하였다. STK 기반의 구조적인 SRS는 내부적으로 제작된 분석 프로그램을 통해 SRS 분석 자동화가 적용될 수 있었다. 실제로 해당 프로그램을 통해 추출된 정보들이 pair-wise와 같은 대표적인 테스트 케이스 생성방법에 적용 가능하였다. 생성된 테스트 케이스는 SRS의 내용을 토대로 실제 무기체계를 모방한 SW를 제작하여 테스트하는데 사용되었다. 해당 무기체계 SRS에 대한 자세한 내용은 보안상의 이유로 생략한다.

2. 작성 보조 도구

본 문단에서는 작성 보조 도구를 이용한 SRS 문장 검사 결과를 소개한다. 먼저 “컴퓨터는 운용자의 요청에 따라 데이터를 처리한다.”라는 문장에 대하여 수행한 검사 결과는 그림 10과 같다. 검사 결과는 해당 문장과 문장에 포함된 모든 안긴 문장을 포함하여 각 문장별로 성공 또는 실패 결과를 표시한다. 성공한 경우, 해당 문장이 속하는 template 번호를 표시하며, 실패한 경우, 오류에 관한 상세 내용이 덧붙여진다. 그림 10의 결과를 통해, 예시 문장은 일반 문장 1에 속한다는 것을 알 수 있다.

또한 검사 결과는 이 외에도 각 문장에 대하여 주어, 서술어, 목적어 등의 문장 성분을 표시하여 각 문장의 구조를 추가적으로 나타낸다. 이 때, 모든 문장의 서술어 어미는 종결 어미인 ‘이다’, ‘하다’, ‘되다’ 등으로만 나타내어 용법에 따른 형태 변화를 고려하지 않는다.

또 다른 문장 “작성 보조 도구로부터 SRS의 신뢰성이 검증된다.”를 검사한 결과, 해당 문장은 부사어, 주어, 서술어 순서를 따르고 있으므로 STK template에 위배된다. 따라서 그림 10의 검사 결과는 표준 문장에 해당하지 않는다는 상세 내용과 함께 실패 결과를 표시하고 있으며, 이러한 메시지를 통해 작성자는 SRS를 적절히 교정할 수 있다.

Ⅵ. 결 론

본 연구를 통해 SRS 작성자는 STK에 따라 구조화된 무기체계 SRS 작성이 가능하고 제안된 새로운 STK를 실제로 적용하는 어려움을 줄이기 위해 작성 보조 도구를 지원한다. 제안된 STK는 국방과학연구소에서 작성된 기존 무기체계 SRS를 근거로 정립되었으며 실제로 작성중인 무기체계 SRS에 적용하여 실효성을 검증하였다.

References

[1] Defense Acquisition Program Administration, “Weapon System SW Development and Management,” Defense Acquisition Program Administration Manual 2018-7, 2018.

[2] Srinivas Nidhra, Jagruthi Dondeti, “Black Box and White Box Testing Techniques-A Literature Review,” International Journal of Embedded Systems and Applications, Vol. 2, No. 2, pp. 29-50, 2012.

[3] M. Ehmer Khan, Garmeena Khan, “A Comparative Study of White Box, Black Box and Grey Box Testing Techniques,” International Journal of Advanced Computer Science and Applications, Vol. 3, No. 6, pp. 12-15, 2012.

[4] Rolf Schwitter, “Controlled Natural Languages for Knowledge Representation,” Proceedings of

the 23rd International Conference on Computational Linguistics, pp. 1113-1121, 2010.
[5] ASD (AeroSpace and Defence industries Association of Europe), Simplified Technical English. Specification, ASD-STE100, No. 7, 2017.
[6] Google, Simplified Technical English [Internet], Avaliable on : https://instrktiv.com/en/simplified-technical-english/.
[7] Junsoo Shin, "KOMORAN", Avaliable on : https://github.com/shin285/KOMORAN, 2016 (in Korean).

Jeonggyu Jang (장 정 규)

He received the B.S. degree in electrical and computer engineering from Ajou University, Suwon, Korea, in 2017. His current research interests include embedded systems design and artificial neural network optimization for embedded systems.
Email: goeka213@ajou.ac.kr

Sanghoon Lee (이 상 훈)

He is currently pursuing the B.S. degree at Ajou University in the Department of Electrical and Computer Engineering. His current research interests include embedded systems design and natural language processing.
Email: can7074@ajou.ac.kr

Hoeseok Yang (양 희 석)

He received the B.S. degree in computer science and engineering and the Ph.D. degree in electrical engineering and computer science from Seoul National University, Seoul, Korea, in 2003 and 2010, respectively. He was a Post-Doctoral Researcher in the D-ITET, ETH Zürich, Zürich, Switzerland from 2010 to 2014. His current research interests include HW/SW codesign, reliability- and temperature-aware optimization/analysis of multi-processor system-on-chip (MPSoC), embedded systems design with non-volatile memories, and deep learning for embedded systems.
Email: hyang@ajou.ac.kr