

무기 획득과정에서의 요구사항 결정 프레임워크

윤하늘*, 옥영호, 왕지남

아주대학교 산업공학과

아주대학교 시스템공학과

아주대학교 시스템공학과

Framework for Determining Requirements in the Weapons systems Acquisition Process

Yun Ha-neul, Ok Young-ho Wang Ji-nam

Ajou University Industrial Engineering

Ajou University System Engineering

Ajou University System Engineering

Abstract : In Weapons acquisition process, if the requirements are not clear, it leads to confusion and conflict in business. In the development of next-generation weapons, to define and derive requirements has to start from decision of operational concept. It is difficult task and has many uncertainties. This paper has goal about development of knowledge acquisition process that derive military operational scenario-based requirements acquisition process and framework that classify and save the derived requirements for next project. That framework can generate new class and category, also suggest that verification of requirements. It help to derive and manage the requirements, and make it easy to grasp the requirements. Accordingly, it can help make military requirement plan and weapon acquisition process.

Key Words : 무기획득, 요구사항, 프레임워크

* 교신저자 : night0114@ajou.ac.kr

1. 서론

1.1 제안 배경

미국 스탠디시 그룹이 2009년도에 기업의 프로젝트 성공요인 및 실패요인을 조사한 결과에 따르면 요구사항과 관련된 사항들이 프로젝트의 성공 여부와 밀접한 관계를 갖는 것을 알 수 있다.

표 1은 프로젝트 성공요인을 10위까지 추려내어 정리한 것으로 유저참여, 요구사항의 명확한 정의, 현실적인 기대 등이 프로젝트를 성공으로 이끄는 주요인으로 포함되어있다.

또한 표 2를 보면 프로젝트를 어렵게 만드는 요인에도 역시 요구사항과 관련된 사항들이 3위 이내에 들어간다는 것을 알 수 있다. 프로젝트를 가장 어렵게 만드는 요인으로 사용자 입력정보의 부족을 들 수 있으며 불완전한 요구사항과 명세, 그리고 요구사항과 명세의 변경이 그 뒤를 이어 프로젝트의 성공 여부를 위협하고 있음을 알 수 있다.

마지막으로 프로젝트 실패요인을 정리한 표 3에서 불완전한 요구사항이 프로젝트를 실패하게 만드는 가장 주요한 요인임을 확인할 수 있으며 사용자의 참여부족 역시 프로젝트를 실패하게 만든다는 것을 알 수 있다. 또한 위의 요인들 보다 그 영향력은 부족하지만 요구사항과 명세의 변경 역시 프로젝트가 실패하는데 많은 영향을 준다는 것을 확인할 수 있다. 이처럼 요구사항은 프로젝트에 많은 영향을 주며 이 요구사항을 잘 관리하는 것이 해당 프로젝트를 성공적으로 이끄는 지름길이라 할 수 있다.

이렇듯 요구사항은 매우 중요하지만 이를 도출하고 명확히 정의하는 것은 굉장히 어려운 일이다. 사용자의 입력정보 부족은 요구사항과 명세를 명확히 하는 것을 방해하며 또한 명확하지 못한 요구사항 및 명세는 사용자가 무엇을 원하는지 알 수 없게 만들어 프로젝트의 진행을 어렵게 만든다. 그리고 이는 프로젝트 중간에 요구사항과 개발 방향을 변경하게 만드는 주요한 원인이 된다. 그렇다면 사용자가 많은 정보를 주면 되지 않는가에 대한 의문이 생길 테지만 보통 사용자들은 자신이 정확히 무엇을

원하는지 잘 모르는 경우가 많으며, 처음에 많은 정보를 제공했다고 하더라도 중간에 새로운 요구사항을 추가하거나 사양을 변경해야하는 경우가 생기게 된다. 이러한 경우 재작업을 하거나 심한 경우 프로젝트를 포기해야하는 경우도 생기게 된다.

<표 1> 프로젝트 성공요인

Project Success Factors	
1. User Involvement	15.9%
2. Executive Management & Support	13.9%
3. Clear Statement of Requirements	13.0%
4. Proper Planning	9.6%
5. Realistic Expectations	8.2%
6. Smaller Project Milestones	7.7%
7. Competent Staff	7.2%
8. Ownership	5.3%
9. Clear Vision & Objectives	2.9%
10. Hard-Working, Focused Staff	2.4%

- Standish report chaos 2009

<표 2> 프로젝트 방해요인

Project Challenged Factors	
1. Lack of User Input	12.8%
2. Incomplete Requirements & Specifications	12.3%
3. Changing Requirements & Specifications	11.8%
4. Lack of Executive Support	7.5%
5. Technology Incompetence	7.0%
6. Lack of Resources	6.4%
7. Unrealistic Expectations	5.9%
8. Unclear Objectives	5.3%
9. Unrealistic Time Frames	4.3%
10. New Technology	3.7%

- Standish report chaos 2009

<표 3> 프로젝트 실패요인

Project Impaired Factors	
1. Incomplete Requirements	13.1%
2. Lack of User Involvement	12.4%
3. Lack of Resources	10.6%
4. Unrealistic Expectations	9.9%
5. Lack of Executive Support	9.3%
6. Changing Requirements & Specifications	8.7%
7. Lack of Planning	8.1%
8. Didn't Need It Any Longer	7.5%
9. Lack of IT Management	6.2%
10. Technology Illiteracy	4.3%

- Standish report chaos 2009

무기획득체계의 경우 무기 하나를 개발하고 획득 하는데 매우 많은 비용을 소요하게 된다. 중어뢰 한 발에 약 9억원 정도의 비용이 들어간다고 하니 그 개발 비용 역시 상당할 것이라 예측할 수 있다. 전체 프로젝트 비용 중 30~50%를 차지하고 있는 것이 재작업 비용이라고 한다. 이 재작업 비용의 70~85%는 요구사항 오류에 의해 소요되는 것이다. 따라서 무기체계 획득 및 개발에서도 요구사항으로 인해 소요되는 비용 역시 많은 부분을 차지하고 있을 것이라 예상할 수 있다. 무기체계는 각 체계에 따라 비슷한 요구사항을 가지게 될 것이다. 이에 관련 정보를 재활용하여 개발 비용과 시간을 대폭 줄인 국내 사례 역시 존재한다. 이에 사용자와 개발자가 요구사항을 도출하고 정의내리는 것을 돕기 위해 요구사항의 프레임워크를 만들어 데이터를 재활용할 것을 제안한다.

2. 본 론

2.1 요구사항이란

Sommerville & Sawyer의 정의를 따르자면, 요구사항이란, 무엇이 구현되어야 하는가에 대한 명세이며 시스템이 어떻게 동작하여야 하는지 또는 시

스템 특징이나 속성들에 대한 설명이다. 또한 요구사항은 시스템 개발 프로세스 상의 제한사항일 수도 있다. IEEE 표준용어집에서는 요구사항을 문제를 해결하거나 목적을 달성하기 위하여 사용자가 필요로 하는 조건 또는 기능이라고 정의하고 있으며, 또한 계약, 표준, 명세 또는 정규화된 문서를 충족하기 위하여 시스템 또는 시스템 컴포넌트가 처리하거나 충족시켜야 하는 조건 또는 기능이라고도 정의하고 있다. 위에서 말한 정의의 조건 또는 기능에 대해 문서로 표현된 것 역시 요구사항이라고 정의 내리고 있다.

요구사항이 무엇인지에 대한 일반적인 정의는 없으나 위의 요구사항에 대한 정의를 종합하고 정리해보면 요구사항이란 사용자가 원하는 목표를 달성하기 위해 필요한 조건 또는 기능의 리스트, 혹은 제한 사항이라고 볼 수 있다.

2.2 모의기반획득

모의기반획득(SBA : Simulation Based Acquisition)이란 M&S(Modeling& Simulation) 기법 및 도구를 이용하여 첨단·복합 무기체계의 획득 및 전략 전술 개발을 위한 일련의 업무들을 모의분석 및 검증을 통해 실행하는 과학적이고 체계적인 획득 활동을 의미한다. 이미 선진국에서는 90년대 초부터 M&S에 의한 무기체계 연구 개발을 적극적으로 추진해왔으며 미 신형 핵잠수함 버지니아호의 경우 개발기간을 1년이나 단축하였고 개발비용을 20%나 절감하는 등의 성과를 달성한 바 있다. 국내 역시 M&S를 1980년대부터 일부 적용해왔으며 1988년도에 시뮬레이션 기반의 설계를 선도적으로 적용한 바 있으나 본격적인 적용은 2000년대 들어와서 설계분야를 중심으로 적용되었다. 제안배경에서도 언급했던 비슷한 종류의 무기 개발 관련 데이터를 재활용하여 개발 비용과 개발 시간을 대폭 줄인 사례가 이 M&S를 적용하여 개발한 수중유도무기 경어뢰 청상어의 개발 사례이다. 청상어의 개발 시 사용한 백상어는 수중유도무기 중어뢰로, 청상어 개발 시 백상어를 개발했을 때 사용한 데이터를

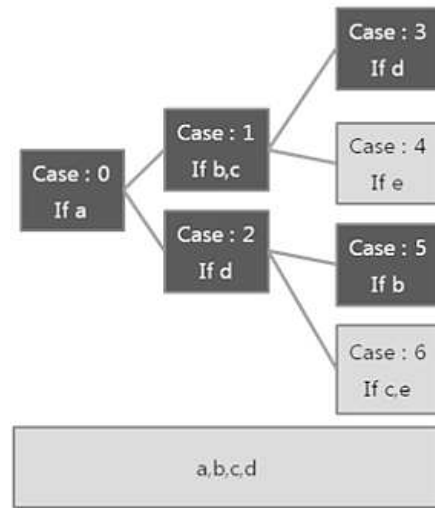
재활용하여 백상어 개발 대비 약 25%의 시간과 비용만이 들었다고 한다. 이를 통해 비슷한 체계의 무기의 요구사항은 각각 비슷할 것이며 데이터 역시 충분히 재활용 가능할 것이라고 예상할 수 있다.

2.3 MCRDR

본 논문에서는 요구사항 도출 및 결정을 도와줄 프레임워크를 구성하기 위해 요구사항을 분류, 저장하는 방법으로써 MCRDR(Multiple Classification Ripple Down Rules)을 활용할 것을 제안한다. MCRDR을 이해하기 위해서는 우선 RDR(Ripple Down Rules) 이론에 대해 먼저 알아둘 필요가 있다. 지식 획득 작업을 위해 지식공학자가 따로 필요했던 기존의 전문가시스템과는 달리 RDR은 지식공학자의 도움을 배제하고 각 분야의 전문가에 의해 지식베이스 시스템을 효율적으로 유지보수 및 구축할 수 있게 해준다. 기존의 지식획득 방법이 하나의 새로운 지식을 학습시킬 때, 지식공학자를 통해 시스템에서 사용할 수 있는 지식으로 변환하는 단계를 거쳤었다면 RDR은 전문가가 직접 실시간으로 새로운 지식을 입력함으로써 보다 빠르게 지식의 변화에 대응할 수 있는 것이다. 그러나 RDR은 오직 하나의 결론만을 만들어 낼 수 있다는 단점을 가진다. 예를 들어 여러 질병에 동시에 걸렸을 때 각 증세들을 가지고 RDR을 수행하여 병명을 진단하는 경우, 모든 증세를 복합적으로 나타내는 하나의 병으로 진단하게 된다. MCRDR은 이러한 RDR의 단점을 보완하여 여러 개의 독립적인 결론을 다룰 수 있도록 한 것으로 MCRDR의 지식베이스 구조는 규칙(rule)로 표현하는 각 노드(node)들로 구성된다. 먼저 첫 번째 레벨의 규칙들을 평가한 뒤 그 하위 레벨로 이동하여 규칙들을 평가하게 되며 케이스에 의해 만족되는 규칙이 더 이상 없거나 더 이상 평가할 하위 노드가 없는 경우 동작을 멈추게 된다. 이런 식으로 규칙들을 평가하여 여러 경로를 가진 복수결론을 도출해낼 수 있다. 예를 들어 그림1과 같은 지식베이스 구조에 a,b,c,d라는 조건을 가지고 있는 데이터를 MCRDR을 통해 분류하게 되면 케이

스 0, 1, 2, 3, 5라는 결과를 얻게 된다. 가장 먼저 케이스 0의 조건을 만족하는지 체크를 하게 되고 해당 조건을 만족하게 되면 그 하위 케이스의 조건을 만족하는지 차례대로 체크하게 된다. 이를 반복하여 최종적으로 위와 같이 케이스 0, 1, 2, 3, 5라는 결과물을 사용자에게 반환하게 되는 것이다. 사용자는 적합하다 생각되는 케이스를 선택하여 해당 데이터를 저장할 수 있으며 또는 새로운 규칙을 추가하여 새로운 케이스를 만들 수도 있다. 지식공학자의 도움 없이 실시간으로 지식획득이 가능하다는 측면에서 기계학습이론보다 우수하며 복수 결론을 다룰 수 있어 RDR보다 우수하다고 볼 수 있다. 본 연구에서는 이러한 MCRDR을 활용하여 요구사항들을 분류 및 저장하는 지식베이스 시스템을 구성하고자 한다.

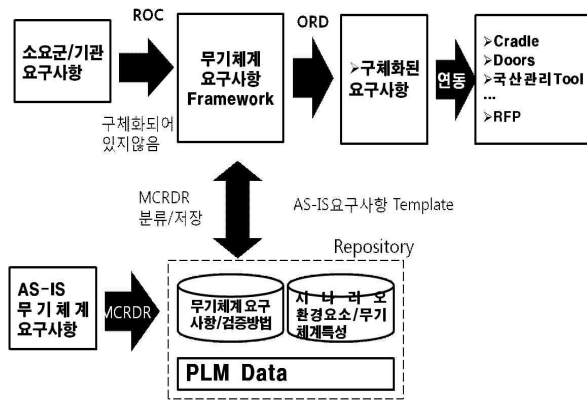
2.4 프레임워크



[그림 1] MCRDR의 예

특정한 상황에 처했을 때 효율적으로 대처하기 위해서 군에서는 작전 시나리오를 작성한다. 작전 시나리오는 해당 작전이 수행되는 지역의 특성과 작전 목표를 고려하여 작성되며 작전 시나리오대로 작전을 수행하기 위해 군에서는 특정한 특성을 갖는 무기를 요구하게 된다. 작전이 수행되는 지역과 그 작전의 특성에 따라 요구되는 무기의 특성 및 종류가 달라지게 되며 이것들 역시 무기의 요구사항

이라고 볼 수 있다. 그렇기 때문에 무기를 획득하거나 개발하고자 할 때 이를 활용하여 획득한 무기를 활용하고자 하는 상황과 가장 흡사한 작전 시나리오의 조건들을 바탕으로 기존에 구축해둔 요구사항 라이브러리에서 관련 요구사항을 뽑아내어 사용자에게 제공하는 것이 가능할 것으로 보인다.



[그림 2] 프레임워크 구조

그림 2는 프레임워크의 구조를 그림으로 표현한 것으로 우선 소요군/기관에서 소요제안/소요요청/소요제기를 거쳐 군요구성능(작전운용요구성능, ROC : Requirement of Capability) 문서를 작성하게 되면 위에 설명한 MCRDR을 활용하여 각 요구사항 및 관련 정보들을 클래스로 정리하여 구성된 지식베이스에서 운용요구서(ORD : Operational Requirement Document)를 작성하기 위한 정보들을 뽑아내어 요구사항을 구체화하는 것을 도와주는 형태로 구성되어 있다. 군요구성능은 구입하고자하는 무기의 성능범위를 정하는 것을 말하며 이것이 결정되면 국방부에서 예산을 마련하여 사업을 추진하게 된다. 운용요구서는 대상 시스템에 대한 소요를 분명하게 제시해주기 위한 것으로 이것이 작성되지 않을 경우 획득기관에서 작성된 체계규격서는 근거가 미약하여 계약문서의 역할이 어렵게 되며, 이 때문에 사용자 요구사항을 근거로 체계 요구사항을 도출하는 것이 일반적이다. 이렇게 도출된 운용요구서는 Cradle이나 DOORS, 혹은 국산 관리 Tool들과 제안요청서(RFP : Request for Proposal)들과 연동되

어 활용된다.

지식베이스에는 무기체계 요구사항과 그 검증방법을 저장할 뿐만 아니라 관련된 시나리오 환경요소 및 무기체계 특성에 대한 정보를 저장하게 되어 추후 군요구성능 문서와 작전 시나리오를 받아 요구사항을 구체화할 때 관련 정보를 제공받을 수 있게 한다. 이후 무기 개발 및 소요 계획을 세울 때 군요구성능 문서와 작전 시나리오를 참고하여 시나리오에서 뽑아낸 특성들과 관련된 요구사항 클래스를 찾아 그 클래스 안의 리스트를 뽑아 사용자에게 제공하는 것이다. 먼저 각 요구사항들을 무기별로 분류하여 개발 및 획득하고자 하는 무기별로 원하는 요구사항을 쉽게 찾을 수 있도록 분류하고자 하며 무기별로 분류한 요구사항들은 MCRDR을 활용하여 특정기능별로 분류하고자 한다. 이는 작전 시나리오 특성과 관련된 기능별로 요구사항들을 제공하기 위해서이다. 이를 통해 사용자는 원하는 기능과 관련된 요구사항 리스트를 받아볼 수 있게 되며 원하는 요구사항을 선택하여 적용하거나 혹은 기존의 요구사항을 기반으로 새로운 요구사항을 도출하여 적용하는 것이 가능해진다. 또한 기존 요구사항과 다른 새로운 요구사항을 사용하게 되는 경우, MCRDR을 활용하여 기존의 클래스 내에 추가하거나 새로운 클래스(rule)로 추가할 수 있다. 해당 프레임워크를 구축하는 것으로 요구사항을 보다 더 쉽게 도출하고 결정할 수 있으며 실제로 개발 및 소요계획에 사용되었던 요구사항들로 구성되어있기 때문에 해당 소요계획을 진행하면서 변경 및 수정을 해야 할 가능성 또한 낮아지게 되어 요구사항 변경 및 수정으로 인한 비용을 절감할 수 있을 것으로 예상된다. 또한 프레임워크의 지식베이스에 이전에 사용했던 요구사항과 관련된 다른 정보들도 저장함으로써 이전 요구사항을 검증하는데 사용했던 검증방법을 제공하거나 비용 예측에 도움을 주는 등 다른 부가적인 도움 역시 줄 수 있을 것으로 생각된다. 해당 프레임워크의 리스트를 사용하고 난 뒤, 무기체계개발 및 획득 프로젝트의 결과 및 최종 요구사항의 데이터를 다시 프레임워크에 피드백하여

계속해서 지식베이스를 보완 및 수정해 나가며 많은 데이터를 축적하게 된다면 무기 획득 과정에 많은 도움이 될 것으로 예상된다.

3. 결 론

3.1 예상 결과

본 논문에서 제안하는 방법론을 적용하게 된다면 개발 또는 획득하고자 하는 무기체계와 관련된 요구사항 리스트를 요구사항 도출 단계에서 받아보므로써 좀 더 명확하고 상세하게 요구사항을 도출하고 정의할 수 있을 것으로 예상된다. 명확하고 상세하게 요구사항을 도출함으로써 중간에 갑작스런 변경으로 인한 비용 및 시간 증대 역시 줄일 수 있을 것으로 예상되며 또한 이는 경어뢰 개발 사례에서 데이터 재활용을 통해 많은 시간과 비용을 단축하였듯이 무기체계획득에 드는 전체 비용 및 시간을 절감하는 것에도 많은 도움이 될 것이라 예상된다.

3.2 결론 및 향후 연구 방향

요구사항을 명확히 정의하고 활용하는 것은 프로젝트의 성패 측면에서도 매우 중요하지만 비용적 측면에서도 매우 중요한 위치를 차지하고 있다. 그 이유는 요구사항 오류를 정정하는데 들어가는 비용이 전체 프로젝트 비용에서 상당한 부분을 차지하고 있기 때문이다. 그렇기 때문에 요구사항을 도출하고 정의하는 것을 도와주는 것만으로도 많은 비용을 절감할 수 있으리라 예상할 수 있다. 특히 상당한 비용을 요하는 무기 개발 및 획득의 경우, 큰 효과를 볼 수 있으리라 예상되며 이에 본 논문에서는 무기 획득과정에 필요한 요구사항 도출 및 결정을 돕는 프레임워크를 활용할 것을 제안하였다. 해당 프레임워크를 구현하고 적용하는 것으로 요구사항 변경 및 수정으로 인해 발생하는 비용이 많이 절감될 것으로 예상되며 또한 요구사항을 도출 및 결정하는데 드는 시간도 단축할 수 있어 개발 및 획득에 소요되는 시간 역시 단축될 것으로 예상된다. 그러나 프레임워크를 구축하기 위한 템플릿 라이브러

리를 구축하는 과정에서 데이터를 표준화해야할 필요가 있을 것으로 보이며 표준화 데이터를 선별해야할 필요 역시 존재한다. 또한 이 표준화 데이터에 어떤 정보를 포함시킬 것인지에 대한 논의 역시 필요할 것으로 보인다. 앞으로 관련 연구를 진행하면서 해당 프레임워크의 상세한 구조와 실제로 무기 획득과정에 어떻게 적용할 것인지 그 방법에 대해 연구해 볼 예정이다.

참 고 문 헌

1. The Standish Group, "CHAOS Summary 2009", 2009.
2. 강병호, 박덕진, 복수결론을 유도하는 지식획득이론, 한국정보과학회 1998년도 가을 학술발표논문집 25(2), 9-11, 1998.
3. Karl Wieggers, Software Requirements, Microsoft Press, 2003.
4. Karl Wieggers, More about Software Requirements : Thorny issues and Practical Advice, Microsoft Press, 2006.
5. ISO/IEC JTC1/SC7 WG4, Study Group Report - Requirement Engineering Tool Requirements, ISO/IEC JTC1/SC7 WG4, 2003.
6. R. Smith, "Simulation Based Acquisition", ModelBenders LLC
7. "Simulation-Based Acquisition (SBA) - Overview", 국방과학연구소, 2007
8. 김대석, 최광목, "무기체계 획득 패러다임 혁신과 SBA", 국방과 기술, 2월호, 32-38, 2000