

CASysE 도구를 이용한 요구사항 관리 및 분석 프로세스 구현 방안

Implementation of Requirement Management and Analysis Process using CASysE Tools

박철호* (아주대학교), 이중윤(에스이테크놀로지), 박영원(아주대학교), 지철규(국방과학연구소), 류태규(국방과학연구소)

1. 서 론

요구사항은 시스템의 존재이유를 정의하기 때문에 명확한 요구사항 아키텍처 구축은 프로젝트의 성패에 매우 결정적인 역할을 한다. 특히 새로운 시스템은 갈수록 다양한 임무를 통합적으로 수행하는 경우가 많으므로 그 구조가 복잡해지고 생명주기에 관련된 이해관계자가 많아지고 있다. 이러한 경향을 잘 나타내는 분야 중 하나가 항공기의 개발이다. 현재 진행 중이거나 최근 개발이 완료된 세계의 군용 항공기 개발 프로젝트를 살펴보면 초기의 개발일정 및 비용으로 성공적으로 프로젝트가 종료되는 사례가 거의 없을 정도로 개발비용 및 일정의 증가가 큰 문제점이 되고 있다.

시스템이 성공적으로 개발되기 위해서는 이에 관련된 모든 이해관계자의 요구가 체계적으로 도출/관리되어야 한다. 그리고 개발기간 동안 발생할 수 있는 요구사항의 변경이 신속하고 일관되게 적용될 수 있도록 복잡한 시스템의 계층구조에 따라 모든 요구사항들의 추적성이 확보되어 프로젝트 관리자의 신속하고 정확한 판단에 도움이 되는 요구사항 데이터베이스의 확보가 필요하다.

미국의 Electric Industries Alliance에서는 시스템공학의 표준 프로세스로서 EIA-632를 제정하였다. 이는 시스템공학의 최초 표준인 MIL-STD-499의 정통성을 인정받고 있으며 국제 시스템 공학 협회인 INCOSE에서는 EIA-632의 How To를 지원하는 내용으로 Systems Engineering Handbook을 발간하였다. 즉 사실상 INCOSE에서 가장 영향력있고 표준적인 시스템공학 표준으로 인정한 것으로 볼 수 있다.

본 논문에서는 EIA-632에서 제안하는 프로세스를 기반으로 하여 시스템 개발에 적용할 수 있는 요구사항 관리 및 분석 프로세스를 제시하였으며 이 프로세스를 상용 시스템엔지니어링 도구(CORE) 및 상용 요구사항 관리 도구(DOORS)를 사용하여 구현할 수 있도록 구체적인 사례를 제시하였다.)

2. EIA-632 프로세스와 요구사항 정의

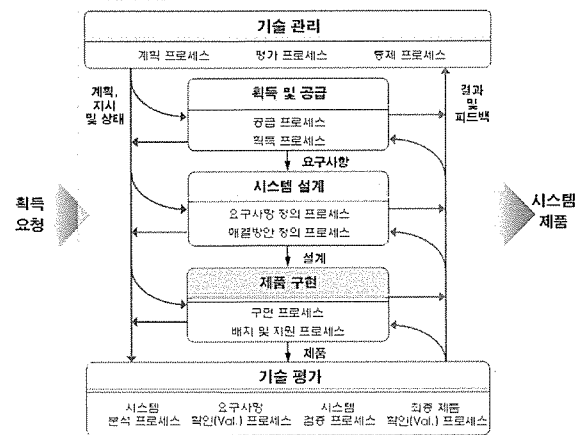


그림 1 EIA-632의 시스템 공학 프로세스

EIA-632 표준 문서는 시스템공학을 수행하는 영역과 이에 대한 거시적인 프로세스를 제시하고 각 프로세스에서 수행하여야 할 업무에 대한 요구사항을 명시하였다. 그림 1은 EIA-632에서 제시한 시스템공학 프로세스이다.

그림 1의 프로세스는 크게 기술 관리, 획득과 공급, 시스템 설계, 제품 구현, 기술 평가 등 5개의 영역으로 구분되어 있으며 이중 시스템 설계 영역에서 요구사항 정의 프로세스를 포함하고 있다. 시스템 설계 영역의 프로세스는 그림 2와 같이 정의되었다.

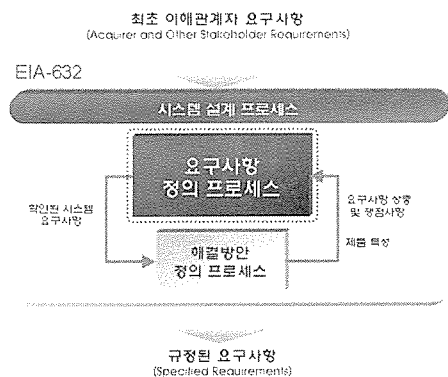


그림 2 EIA-632의 시스템 설계 영역

그림 2의 시스템 설계 프로세스에 따르면 우선 그전에 존재하는 획득과 공급 프로세스에 의해 시스템 개발이 결정되고 최초 이해관계자 요구사항이 발생되면 그에 의해 시스템 설계 프로세스가 시작된다. 시스템 설계 프로세스는 “요구사항 정의 프로세스”와 “해결방안 정의 프로세스”로 구성된다. 요구사항 정의 프로세스에서는 시스템 개발 시 고려해야할 요구사항들을 도출, 수집, 확인(validate)하게 되고 “요구사항 정의 프로세스”의 산출물은 “확인된 시스템 요구사항(Validated System Requirements)”이 된다.



그림 3 EIA-632의 기술 평가 영역

EIA-632 프로세스의 “기술 평가”영역(그림 3)에 정의되어 있는 “요구사항 확인 프로세스”에서는 “확인된 시스템 요구사항(Validated System Requirements)”이 갖추어야 할 요건들에 대한 정의를 하고 있다.

이 정의는 EIA-632 표준 문서의 부록 C에 있는 표 C.25에 정리되어 있으며, 크게 요구사항 문장이 독립적으로 갖추어야 하는 속성 11개와 요구사항의 집합 속에서 요구사항 문장들이 갖추어야 하는 속성 3개로 구분된다. EIA-632에서는 이러한 요구사항이 갖추어야 할 속성에 대한 나열과 간략한 설명은 되어 있지만 특정 시스템 도메인의 영향을 받을 수 있는 세부 사항에 대한 언급을 회피하는 표준 문서의 특성상 어떻게 수행하여야 할지에 대한 구체적인 방법은 제시하고 있지 않다.

3. 기 연구된 요구사항 정의 프로세스

위에 소개된 EIA-632의 요구사항 정의 프로세스에 대한 구체적인 수행방법이 2004년 아주대학교의 이중윤에 의해 연구된 바 있다. 이 연구는 초기 요구사항 도출을 효율적으로 수행하기 위한 템플릿의 제안과 요구사항 정의 프로세스에 대한 세부 프로세스를 제시하고 각 프로세스에 대한 활동사항을 정의하였다.

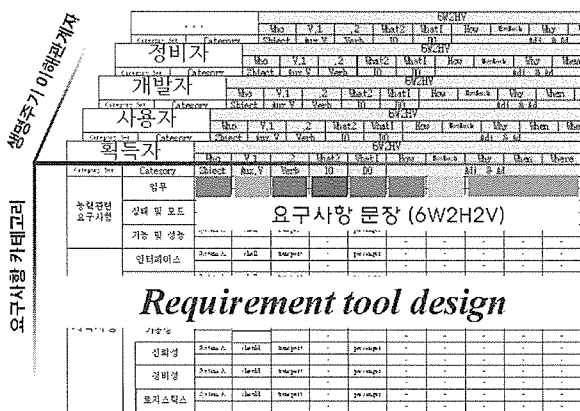


그림 4 Requirement Architecture Framework

우선 RAF(Requirement Architecture Framework)로 명명된 이 템플릿은 가로축이 요구사항을 더욱 명확히 기술하기 위한 6W2H2V(누가, 언제, 어디서, 어떻게, 왜, 무엇을, 누구에게, 얼마나. 동사 및 조동사) 항목으로 구성되어 있고 세로축은 임무, 기능, 인터페이스 등 요구사항의 범주를 제시하고 있다.

가로축의 6W2H2V는 초기 요구사항을 도출할 때 내용상의 무결성을 높이기 위한 것이고, 세로축의 요구사항 범주는 요구사항의 도출자가 특정 범주에 치우치지 않고 요구사항 도출시 고려해야 할 분야를 제시하여 누락되는 요구사항을 최소화하고자 함이다. 그리고 시스템의 생명주기에 걸친 모든 이해관계자에 대하여 위 템플릿을 작성하는 것이 RAF의 개념이다.

그림 5는 EIA-632에서 정의한 요구사항 정의 프로세스를 수행하기 위해 더욱 구체적으로 정의한 것이다.

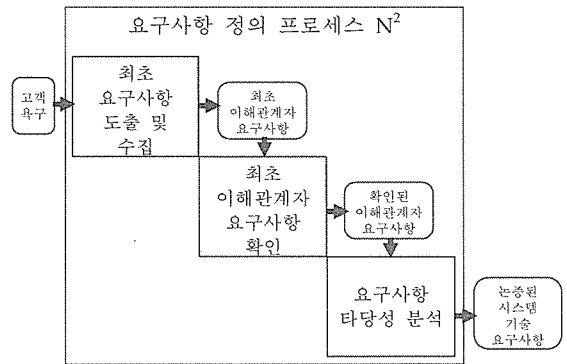


그림 5 이중윤의 시스템 요구사항 정의 프로세스

우선 시스템의 생명주기 전반의 이해관계자가 고려된 가능한 한 모든 요구사항을 도출하기 위한 프로세스가 선행되고 이렇게 도출된 요구사항의 표현과 의미에 대한 확인이 필요하며 마지막으로 요구사항들의 내용상 타당성을 확인하는 프로세스를 제안하고 있다.

그림 6은 요구사항 정의 프로세스에서 특히 최초 이해관계자 요구사항 확인 프로세스를 좀더 자세히 정의한 것이다. 초기 이해관계자로부터 도출된 요구사항들은 시스템이 아닌 다른 레벨에 대한 기술을 하고 있을 가능성이 높기 때문에 우선 시스템에 대한 기술을 하도록 정리하는 “충별화”작업이 선행되고 차후 요구사항을 시스템 규격서의 범주에 맞게 정리하며 EIA-632가 제시하는 요구사항의 요건에 대해 문장 및 요구사항 집합 확인 절차를 거치는 것을 제안하고 있다.

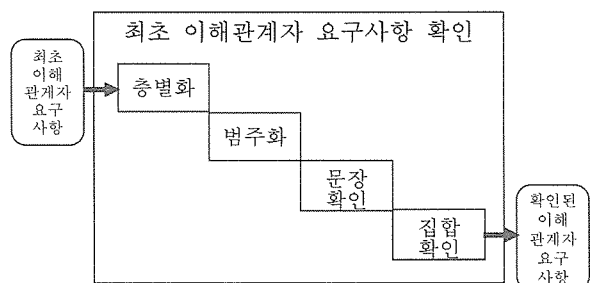


그림 6 최초 이해관계자 요구사항 확인 프로세스

4. 본 연구에서 제안하는 요구사항 정의 프로세스 및 구현 방안

(1) 요구사항 정의 프로세스의 단계별 특징 및 도구와의 관계

이중윤가 제안하는 “시스템 요구사항 정의 프로세스”에서 제시된 3개의 각 프로세스 영역별 특징과 이에 적절한 도구는 표 1과 같은 것으로 판단된다.

프로세스	수행업무의 특징	추천되는 도구
최초 요구사항 도출 및 수집	다양한 이해관계자들과의 데이터 교환 및 정리가 수반됨	일반적으로 널리 사용되는 워드프로세서나 스프레드시트
최초 이해관계자 요구사항 확인	다양한 분야와 많은 요구사항의 정렬, 요구사항의 표현과 내용에 대한 검토 작업 및 그에 따른 각종 이력 관리가 필요함	요구사항 관리를 지원하는 전문 CASE 도구
요구사항 타당성 분석	수학적 혹은 기타 전문공학 영역의 모델을 이용한 요구사항 내용의 논리적 검증이 요구됨	기능분석이 지원되는 CASE 도구, 혹은 전문공학에 특화된 분석 도구

표 1 시스템 요구사항 정의 프로세스에 등장하는 하부 프로세스의 특징

본 연구에서는 “최초 요구사항 도출 및 수집”프로세스에 Microsoft Excel을 사용하였으며 “최초 이해관계자 요구사항 확인”프로세스에는 Telelogic DOORS를 사용하였다. 마지막으로 “요구사항 타당성 분석”프로세스는 해당 전문공학 영역별로 다양한 도구가 사용될 수 있으나 이 연구에서는 시스템 공학 통합 도구인 Vitech의 CORE에서 제공하는 기능분석 기능을 이용하여 시스템 수준의 기능분석 사례를 구현해보았다.

(2) DOORS를 이용한 요구사항 데이터베이스 구현 방안

Telelogic사의 DOORS는 대표적인 요구사항 관리 도구이며 네트워크 기반의 작업을 수행할 수 있는 다중 사용자 데이터베이스이다. DOORS의 데이터 구조는 folder, project, formal module로 구성되어 있다. folder와 project는 독립된 저장 공간의 개념이며 이 중에서 folder는 시스템에 대한 외부적인 제약사항이나 ROC와 같은 잘 변하지 않는 내용들이 위치하며 과제에 대한 대부분의 요구사항은 project 하에 위치하게 된다. formal module은 요구사항이 저장하고 있는 단위체이다.

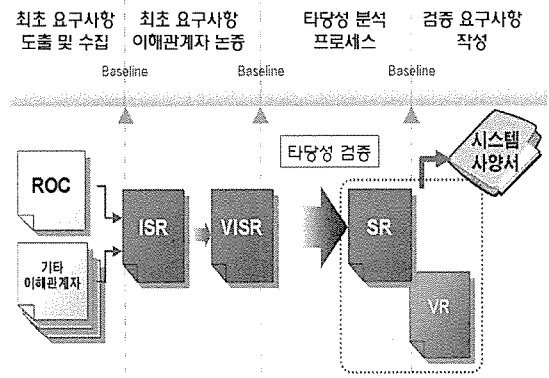


그림 7 시스템 정의 프로세스를 위한 DOORS 운용 스키마

본 연구에서 최종 제안하는 시스템 요구사항 정의 프로세스를 수행하기 위한 DOORS 운용 스키마는 그림 7과 같다.

그림 7에 제시된 스키마는 ROC를 비롯한 초기 요구사항들이 독립된 formal module로 작성되는 것을 나타낸다. 그리고 초기 요구사항들은 ISR 모듈로 통합되어 요구사항 확인 프로세스를 거친 후 모듈명을 VISR(Validated Initial Stakeholder Requirement)로 수정하게 된다. VISR은 타당성 분석 프로세스를 거친 후 시스템의 기술적 요구사항인 SR(System Requirements)로 수정된다. 그리고 각 요구사항에 대한 검증 요구사항을 작성하고 SR과 VR을 출력하여 통합하면 시스템 규격서가 되는 형상을 취하고 있다.

본 연구에서는 앞에서 정의된 시스템 요구사항 정의 프로세스를 실제로 DOORS를 이용하여 구현을 하기 위한 세부 수행 방법을 여러 가지 설정한 다음 요구사항 집합을 임의로 가정하여 직접 수행하였다. 다음에 그 결과를 기술하였다.

a. ISR 모듈의 단일화

ISR(Initial Stakeholder Requirements)는 모든 초기 이해관계자로부터 수집된 초기 요구사항들이 통합되어있는 요구사항 집합이며, 요구사항 확인 프로세스가 수행되는 모듈이다. 연구 초기에는 요구사항 확인 프로세스의 4단계 내부 프로세스 별로 각각의 모듈을 작성하는 것이 고려되었다. 이는 4단계 프로세스별 활동내역과 그에 따른 산출물을 구별하고 관리하기 위한 마일스톤(milestone)을 잡기 위한 것이었다. 그러나 그에 따른 DOORS 데이터베이스의 formal module 증가와 그 formal module에 존재하는 요구사항들 간의 추적성으로 인한 데이터베이스의 복잡도 증가가 문제가 되었다. 따라서 요구사항 확인 프로세스의 4단계를 ISR 모듈내에서 모두 작업하며, 단계별 결과물 관리 및 추적은 DOORS에서 제공하는 “history” 및 “baseline” 기능을 이용하여 해결할 수 있었다.

b. 시스템 요구사항과 검증 요구사항의 분리

본 연구에서 목표로 설정한 시스템 규격서 규격

은 MIL-STD-961D이었다. 이 규격은 많은 다른 경우와 마찬가지로 규격서내에 요구사항과 그에 대응하는 검증 요구사항이 기재되도록 되어 있다. 연구 초기에는 DOORS가 하나의 formal module 내에서 링크 생성을 허용하기 때문에 시스템 규격서와 대응하는 formal module을 생성하도록 하였으나, 이렇게 되면 나중에 문서의 크기가 커졌을 때 추적성 관리 작업이 용이하지 않았다. 따라서 시스템 요구사항(SR)과 검증 요구사항(VR)을 분리하였다. 후일 시스템 규격서를 출력할 때에 약간의 번거로움은 있으나 전체적인 작업 효율은 증가하였다.

c. 변경이력관리 시 DOORS의 CPS 기능 활용
DOORS에서는 요구사항의 변경제안이 요구사항에 직접적으로 적용되는 것을 통제하고 변경이력을 관리하기 위해 CPS(Change Proposal System)이 제공된다. 연구 초기에는 발생하는 변경제안을 독립된 formal module에 정리하고 요구사항과의 추적성을 관리하는 요구사항의 이력관리 방법을 추진하였다. 이는 요구사항 변경 제안 및 그 결과들을 정리/출력하는 기능을 CPS에서 아직 제공하고 있지 않기 때문이다. 그러나 CPS는 사용자 계정과 권한에 따라 변경제안을 수집/관리하고, 요구사항에 자동으로 적용되며, 변경제안의 발생을 담당자에게 e-mail로 발송하는 등의 부가 기능이 잘 발달되어 있다. 이와 같은 자동화된 기능의 지원은 업무의 효율과 데이터베이스의 일관성 유지에 많은 도움이 되기 때문에 CPS 기능을 활용하였다.

d. 요구사항의 평가와 속성(attribute) 활용
EIA-632 프로세스에서는 요구사항이 갖추어야 할 요건에 대한 기준이 제시되어있고, 이중윤은 이들이 실제로 평가되어야 함을 주장하였다. 본 연구에서는 DOORS 상에서 평가가 어떤 형태로 이루어지는 것이 좋은지에 대한 고찰을 하였다. DOORS는 요구사항에 대한 속성(attribute)을 무제한으로 새롭게 정의할 수 있으며 속성의 데이터 형식 또한 다양하게 지원한다. 따라서 요구사항 확인 프로세스의 4단계에 대한 각 평가기준들의 평가척도를 정립하면 DOORS상에서 속성을 이용하여 평가를 기록할 수 있다. 다수개의 속성에 대한 필터링, 정렬, CPS 기능들이 지원되기 때문에 이와 같은 활동은 매우 효율적인 것으로 나타났다. 그림 8은 DOORS에서 해당 속성을 정의하여 문장확인 작업을 하는 하나의 사례 화면이다.

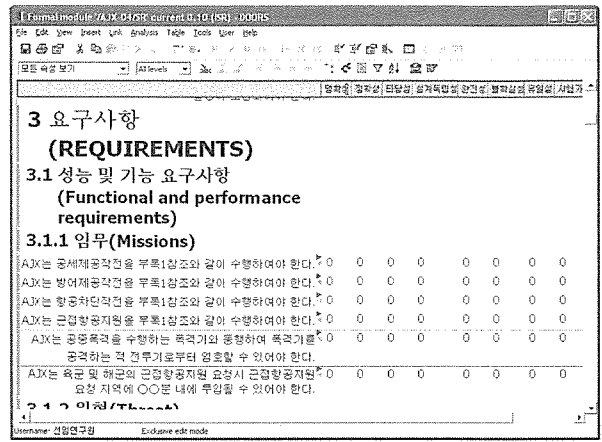


그림 8 DOORS에서의 문장확인 평가 화면

(3) CORE의 기능분석을 이용한 타당성 분석 방안

EIA-632는 요구사항이 갖추어야 할 요건 중 하나로 타당성을 제시하고 있다. 타당성은 다른 요건과는 달리 요구사항 단일 문장이나 요구사항 집합 속에서 이루어지는 표현적인 판단이 아니라 논리적, 물리적 사실에 기초하여 전문적이고 분석적인 접근이 필요하다. 따라서 본 연구의 프로세스에서는 타당성 분석을 요구사항 확인 프로세스 이후에 수행하는 독립적인 프로세스로 제안하였다.

타당성 분석을 수행하는 방법은 기본적으로 여러 물리적, 논리적 양성을 나타내는 모델을 수립하고 주어진 환경변수에 대한 결과를 도출하여 요구사항의 내용이 주어진 상황 및 일정, 비용적인 측면에서 적절한 것인지를 판단하는 것이다. 따라서 각종 전문공학에서 모든 해석적 모델이 적용될 가능성이 있다. 본 연구에서는 일반적인 통합 CASE 도구에서 지원하는 기능분석을 이용한 시스템의 타당성 분석 방법에 대한 연구를 수행하였다.

획득자가 항공기의 소요를 제기하는 이유는 사용하고자 하는 목적이 있기 때문이며 군용 항공기의 경우, 이러한 목적은 곧 작전이 된다. 작전은 하나 이상의 임무로 구성되어 있고 개개 임무의 수행절차를 기술한 것이 임무 프로파일이다.

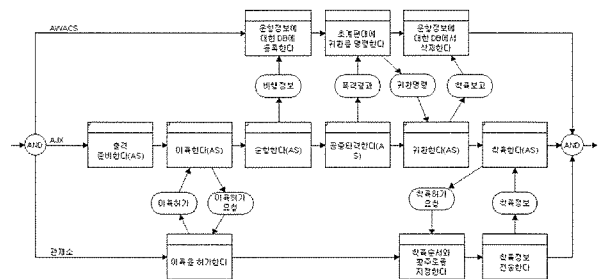


그림 9 타격임무의 EFFBD

그림 9는 가상의 타격임무에 대한 임무 프로파일을 Vitech의 CORE를 이용하여 EFFBD로 구현한 것이다.

(본 논문에 등장하는 모든 다이어그램과 특정 숫자는 기능분석의 구현 방법에 대한 사례 연구를 위해 가정한 것이며, 특정 상황과는 무관하다.)

그림 9에서 가운데 줄에 위치한 기능이 시스템에 관한 것이며 그 밖의 것들은 임무 수행간 시스템과 상호작용을 하는 외부 시스템이다. 최상위의 EFFBD에는 반드시 시스템과 관련이 있는 외부 시스템이 등장하여 기능의 흐름과 데이터 연계성을 모델에 구현해야 한다.

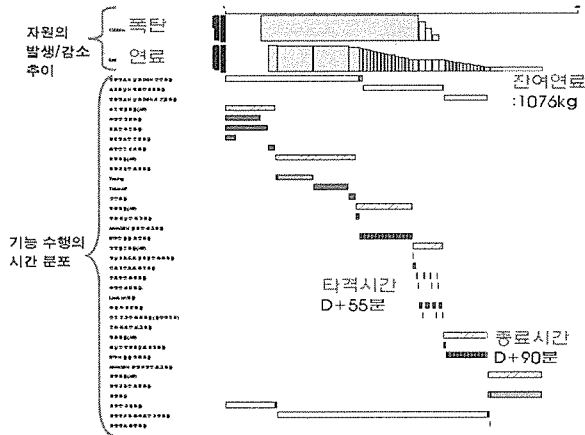


그림 10 시뮬레이션을 통한 자원 소모 분석

시스템의 기능 혹은 거동들이 정해지면 모델의 분해가 필요하다. 분해의 수준은 모델링의 목적에 따라 상이하다. 본 사례에서는 타당성 분석의 사례로 시스템의 연료와 무장에 대한 소요량의 분석을 택하였다. 따라서 EFFBD는 연료나 무장의 소요량을 할당할 수 있는 수준으로 분해되어야 한다.

EFFBD가 필요 수준으로 분해되면 무장 및 연료 등의 자원을 소모하거나 생성하는 기능에 해당 자원을 할당한다. 여기서 자원을 할당하기 전 해당 임무에 대한 경계조건 설정이 필요하다. 상황에 따라 소요되거나 생성되는 자원에 대해 일반화된 수학적 모델을 기능에 할당할 수도 있다.

5. 결론

본 연구는 요구사항 분석 및 관리에 대해 EIA-632에서 제시하는 시스템공학 프로세스를 기반으로 하였으며, 이중윤이 그의 논문에서 제안하였던 프로세스와 방법론을 활용하여 CASysE 도구 상에서 어떤 방법으로 구현하는가에 대한 연구를 수행하였다. 즉 James N. Martin이 주장한 PMTE 패러다임(프로세스 ⇨ 방법론 ⇨ 도구 ⇨ 환경)에서 CASysE 도구에 해당하는 영역에 대한 연구이다.

EIA-632에서 제시하는 요구사항 정의 프로세스를 수행하기 위해 Telelogic DOORS의 운용에 관련된 교환과 Vitech CORE를 이용한 기능분석 방안을 제시하였다. 위의 내용들은 요구사항 관리 및 분석 프로세스를 상용 도구를 사용하여 실행 가능한 수준으로 구체화하여 제시하였다는 점에서 유용하다.

6. 참고문헌

1. Electronic Industries Alliance(EIA), EIA 632: Processes for Engineering a System, Electronic Industries Alliance, USA, 1998
2. Martin, J., Systems Engineering Guidebook: A Process for Developing Systems and Products, CRC Press, USA, 1996
3. 이중윤, 시스템 요구사항 정의 프로세스 및 도구 개발, 아주대학교 대학원 박사학위논문, 2004