

SE 프로세스 기반 실시간 시스템 개발에 관한 연구

A Study on the development of real-time systems based on SE process

이병길*, 이재천(아주대학교)

1. 서 론

실시간 시스템(real-time system)은 1950년대 통신장비를 제어하기 위해 컴퓨팅시스템이 전자기기에 내장되면서 등장하기 시작하였다. 1990년대 이전까지는 군사용 기기와 산업용 기기에만 제한적으로 사용되었던 실시간 시스템의 응용 분야가 3C(Computing, Communication, Consumer Electronics)의 기술 융합이 나타난 1990년대 후반부터 급속도로 확대되어가고 있다. 이러한 응용분야의 확산으로 하드웨어와 소프트웨어가 결합된 복잡한 요구사항을 구현해야하는 유비쿼터스 시스템에서의 이용에 관한 연구도 이루어지고 있다. 현재 제시된 실시간 시스템의 개발 방법론으로는 이러한 복잡한 요구사항을 다루기는 쉽지가 않은 것도 사실이다.

현대의 시스템은 하드웨어와 소프트웨어가 상호 보완적으로 결합된 형태가 대부분이다. 그 동안 하드웨어, 소프트웨어의 개발에는 각 전문 영역에 맞는 모델링 언어가 존재하였고, 하드웨어나 소프트웨어 각각의 개발에는 별다른 어려움이 없었다. 하지만 현대 시스템처럼 하드웨어와 소프트웨어가 결합된 형태의 시스템을 모델링하기 위해서는 공통의 모델링 언어의 필요성이 증가되었고, 공통의 모델링 언어로서 제안된 것이 UML(Unified Modeling Language)이다.

이미 연구된 실시간 시스템의 개발방법론은 “무엇을 개발할 것인가?”에 대한 정의는 잘 되어 있지만, 구체적인 방법에 대한 기술은 피하고 있다. 기존의 제안된 방법 중 하나인 ROPES(Rapid Object-Oriented Process for Embedded Systems)에서는 UML을 이용하여 실시간 시스템 개발 프로세스를 표현하고 있다. 하지만 UML을 이용하여 개발프로세스 전 단계에 적용하다 보니 Analysis단계에서 시스템 전체에 대한 고려와 요구사항의 추적성 확보가 어렵다.

본 논문에서는 ROPES의 Analysis단계에서 시스템 엔지니어링 프로세스를 적용하여 시스템 전체를 고려한 개발 방법론을 제시하고자 한다.

본론에서는 먼저 객체지향 실시간 시스템 개발 프로세스인 ROPES에 대해 알아보고, 시스템 엔

지니어링 표준 프로세스중 하나인 EIA632에서 제시한 시스템설계에 대한 프로세스를 기반으로 개선된 실시간 시스템 개발 방법론을 제시하였다.

2. 본 론

2.1 실시간시스템의 개발 프로세스

ROPES 방법의 소개

ROPES는 Bruce Powell Douglass[1] 등이 개발한 실시간 시스템을 개발하는 방법론이다. 이 방법론은 객체지향 방법론을 이용하여 개발한 방법론이다. 이 방법은 프로젝트의 기간과 Spiral방법론을 접합시킨 방법으로 시스템의 개발 기간을 Macro, Micro, Nano 로 나눈다.

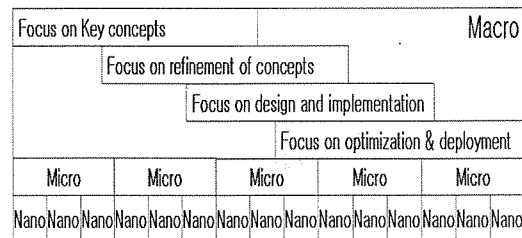


그림 1 . ROPES Timescales

Macro : 프로젝트의 전체적인 기간을 나타낸다. 주기는 4단계(Key Concepts, Secondary Concepts, Design Concepts, Optimization and Development Concept)로 구별된다.

Micro : 시스템의 목표가 되는 요구사항의 기능을 도출하거나 기능의 확장을 효과적으로 한다. 기간은 4~6주 정도로 한다.

Nano : 시스템의 작은 부품을 구성하고, 컴파일하고, 실행 및 테스트하는 단계이다. 기간은 30분에서 1시간 사이로 한다.

ROPES의 각 단계를 살펴보면 다음과 같다.

Party 단계

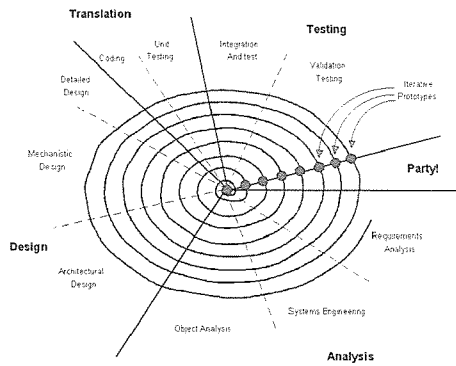


그림 2 ROPES Spiral

Party 단계에서는 과제를 받아 개발범위와 기간, 총 사업 금액을 결정하는 단계로서 대상시스템의 범위와 일정 등을 수립하는 단계이다.

Analysis 단계

Analysis 단계는 3단계로 구성되는데 요구사항을 분석하는 Requirements Analysis 단계, 시스템엔지니어링을 적용하는 Systems Engineering 단계, 객체를 분석하는 Object Analysis 단계로 구성된다. 최초 고객의 요구사항을 가지고, 시스템이 무엇을 하는지에 대한 임무와 시스템 요구사항을 도출하는 단계이다.

Design 단계

Design 단계는 분석된 모델과 구현 기술을 식별하는데 3단계로 구성되는데 Architectural Design, Mechanistic Design, Detailed Design로 나뉘는데, Design 단계의 핵심은 전체 아키텍처 설계를 통해 대상시스템을 설계하고, 대상시스템을 구성하는 구성요소를 상세화시키는 것이다.

Translation 단계

Design 단계를 통해 식별된 기능과 성능을 제약사항에 맞게 코딩하는 단계이다.

Testing 단계

Testing 단계에서는 시제품을 통해 사용자의 요구사항이 잘 반영되었는지 시험하는 단계이다.

2.2 시스템 엔지니어링 프로세스

시스템엔지니어링 프로세스를 설명하기에 앞서 시스템엔지니어링 표준간의 전개과정과 상호 관련성을 이해할 필요가 있다. 시스템엔지니어링 표준은 그림 3에서 알 수 있는 바와 같이 1969년 MIL-STD 499를 토대로 현재와 같이 전개되어 왔다.¹⁾

Halligan²⁾은 현재까지의 모든 시스템엔지니어링 표준은 공통적인 주제를 다루고 있고 매우 유용하지만 너무 많은 기대를 가질 필요는 없다고 주장하였다. 더 나아가 시스템엔지니어링의 정의와 범위에 대한 일반적인 견해가 합의되지 않은 상태이므로 이러한 표준은 무시해도 좋다고 주장하였다. 그가 제시하는 각 표준의 특징을 소개하면 아래와 같다.

- MIL-STD 499B와 499를 제외한 이미 취소되거나 파기된 미국 국방 시스템엔지니어링 표준은 어딘가 문제점을 내포하고 있거나 위험하므로 사용에 주의해야 한다.
- 파기된 EIA/IS 632는 내용이 불충분하지만 유용한 자료를 수록하고 있다. 계약을 위한 적용에는 가장 좋은 표준이다. 현재 사용 중인 EIA 632는 표준이라기보다 하나의 유용한 지침서로써 자발적으로 선택하여 적용할 경우 EIA 632가 일반적으로 좋은 지침서가 된다.
- IEEE 1220은 적용개념이 취약하고 전체적인 방법보다 오히려 세부적인 방법을 제시하고 있다.
- ISO/IEC 15288은 개념적으로 취약한 시스템 수명주기 표준이므로 이를 적용할 경우 조직 측면에서 후퇴가능성 있다.

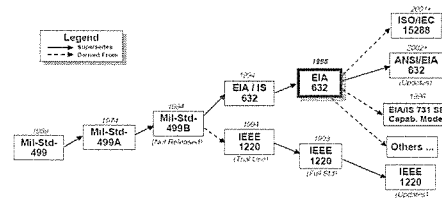


그림 3 시스템엔지니어링 표준의 전개도

2.3 요구사항 템플릿(Template)

Real-Time System과 같은 복잡한 요구사항을 시스템적인 관점에서 접근하기는 쉽지 않다. 어떠한 시스템이라도 이해당사자 및 개발 관련자들의 정보를 누락 없이 정확하게 수집하여야 데이터베이스 구축이 가능하게 된다. 이렇게 요구사항에 대한 정보를 쉽게 수집하고 관리를 용이하게 하기 위해 템플릿은 사용된다. 또한, 기존에 형성된 데이터베이스가 없고, 문서로만 남아 있는 경우에는 문서의 정보를 데이터베이스로 옮겨야 한다. 이 경우에도 입력 템플릿에 따라 문서정보를 정리함으로써 빠르고 정확하게 데이터베이스화할 수 있다. 요구사항 템플릿(RAF: Requirements Architecture

1) Ken Rigby, "Managing Standard V4.4", Proceedings of the 8th Annual INCOSE Symposium, pp. 589-596, 1998.
 2) Halligan, R., "Systems Engineering", Seminar/Workshop Material, April, 2002

Frameworks)의 특징은 요구사항을 6W2H2V로 나누어 기술하는 방법이다. 학위논문3)에서 RAF 템플릿의 사용 목적은 시스템 개발초기에 고객의 요구사항을 더욱 효과적으로 도출하기 위하여 개발하였다.

8하원칙							2동사		
주어	간접목적어	직접목적어	서술어				동사		
누가	누구에게	무엇을	언제	어디서	어떻게	얼마나	왜	동사	조동사
(Who)	(What2)	(What1)	(When)	(Where)	(How)	(How Much)	(Why)	(Verb)	(Aux ^o V)

표 1 RAF 템플릿 형식

실시간 시스템 개발 방법론에 UML 적용의 문제점

실시간 시스템 개발 방법론은 각 단계마다 해야하는 업무를 잘 정의하고 있다. UML을 사용하여 실시간 시스템 개발 방법론을 모든 단계에 적용하는 것에는 문제점이 존재한다.

- UML은 요구사항의 계층구조를 나타내기에 부족하다.
- UML을 통해 Systems Engineering의 단계에서 어떻게 구현할 것인가?
- UML을 통해 사용자 중심의 기능 중심 요구사항이 고려됨.
- 체계적인 절차를 세우기가 어렵다.

위에서 식별된 문제점의 요구사항을 어떻게 관리하고, 기능 중심의 요구사항이 식별된다. 이는UML에서 사용하는 Software 개발절차를 실시간 개발 프로세스에 억지로 적용한 결과이다. 실시간 시스템의 특징은 제한된 CPU를 이용하여 소프트웨어를 구동한다는 점이다. 즉 초기 요구사항 식별 단계부터 하드웨어를 고려한 소프트웨어를 개발해야한다. 하지만 기존의 기능 중심의 요구사항을 통해 실시간 시스템을 개발한다면 현장에서 적용시 설계 변경이 일어날 수 밖에 없다. 기능 중심의 요구사항이 필요하지만, Requirement Analysis 단계에서 하드웨어의 제약사항, 지원, 유지보수, 인터페이스 등의 요구사항이 식별이 필요하다.

전 수명주기 단계를 고려한 요구사항 도출과 요구사항의 추적성의 문제점을 개선하기위해 실시간 시스템의 개발 방법론의 Analysis 단계에 시스템 엔지니어링에서 사용하는 방법을 적용하였다.

2.4 실시간 시스템 개발방법의 Analysis 단계에서 시스템엔지니어링 프로세스 적용

기존 실시간 개발방법에서 기존의 Analysis 단계에서

3) 이중운, “시스템 요구사항 정의의 프로세스 및 도구 개발”, 아주대학교 박사논문, 2004.

시스템 전체를 고려하기위해, 시스템 엔지니어링의 개발 방법론을 적용하였다. 시스템 엔지니어링 프로세스의 특징은 시스템 개발단계에서 시스템의 환경적인 요소, 인터페이스, 비용, 안전제약사항, 시스템의 유지 보수 등의 시스템의 전체적인 관점이 고려된다. 시스템엔지니어링 방법론은 시스템 설계변경을 줄이고, 비용이나, 시간을 단축시킬 수 있다. 시스템엔지니어링 방법론은 국제 표준을 상세화하여 사용하기 때문에 절차를 논증할 수 있다.

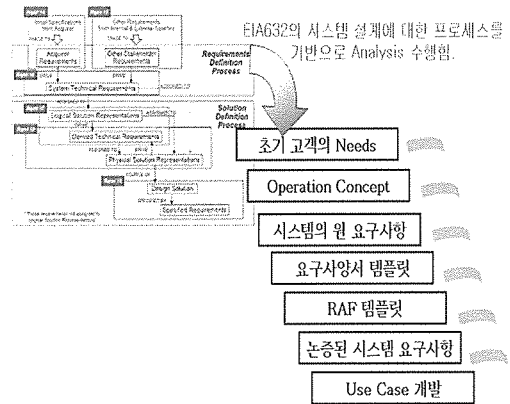


그림 4 Analysis 수행 절차

그림 4는 EIA632의 시스템 설계에 대한 프로세스를 기반으로 Analysis단계를 수행 절차를 표현한 그림이다.

Requirement Analysis단계의 수행 절차를 설명하면 실시간 시스템 개발 초기에 고객의 Needs를 통해 시스템의 운용 시나리오를 개발하였고, 시스템의 운용 시나리오를 통해 시스템의 경계를 설정하고, 시스템 운용 시나리오를 통해 시스템의 원 요구사항을 개발하였고, 시스템 원 요구사항을 시스템 요구사항서 템플릿 항목에 맞게 개발하였다. 모델과의 추적성 확보를 위해 시스템 요구사항서를 RAF의 항목으로 개발하였다. 논증된 시스템 요구사항을 가지고 시나리오 단위로 통합하는 기능이 추가된 RAF 항목의 상태 및 모드, 기능 및 성능 요구사항을 이용하여 시나리오를 구성한다. 각 시나리오를 통해 UML의 Use Case Diagram으로 개발하는 방법을 개발하였다.

5. 개선된 실시간 개발 방법론에 대상적용

시스템엔지니어링 프로세스를 통한 실시간 시스템을 개발하기위하여 이 기중간의 카메라와 제어기를 연결하는 프로토콜 변환기에 적용하였다.

Analysis 단계에서 최초 요구사항 수집단계를 통해 사용자의 Needs를 식별하고, 시스템의 운용시나리오를 통해 시스템의 경계를 정한다. 시스템의 운용 시나리오

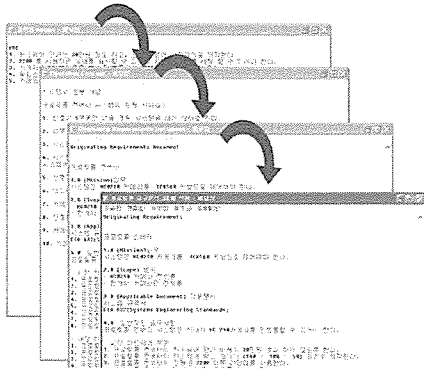


그림 5 Analysis 수행 절차

를 통해 시스템의 임무를 정하였고, 결과물을 이용하여 시스템 요구사항서를 개발하였다. 시스템 요구사항서를 통해 개발된 요구사항은 시스템의 전 수명단계를 지원하는 항목으로 구성되어있기 때문에 시스템의 추상성을 가시화시켜준다. 이 결과물을 RAF에 적용하여 요구사항을 항목별로 구분하였다. RAF적용시 기존요구사항에서 식별하기 어려웠던 시스템의 제약사항, 성능 요구사항이 식별되었다.

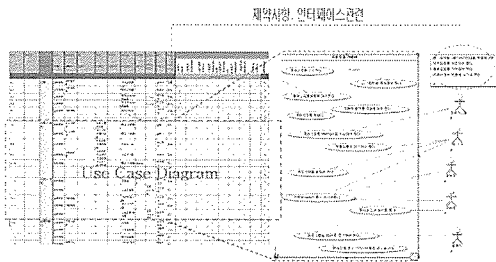


그림 6 템플릿 항목을 통한 Use Case Diagram 모델링

그림 6은 요구사항 템플릿의 항목을 이용하여 Use Case Diagram을 모델링 한 것이다. 제약사항, 인터페이스와 관련된 요구사항은 메모장이나 description을 이용하여 시스템을 개발하는데 고려할 수 있도록 개발하였다.

III. 결론

본 논문에서는 실시간 시스템의 개발프로세스의 Analysis 단계에서 기존에 고려되었던 기능 중심의 요구사항을 시스템엔지니어링 개발 프로세스를 적용하여 시스템의 전 수명주기가 고려된 시스템 요구사항을 개발하였다.

본 연구를 통하여 기여하는 바는 다음과 같다.

- Analysis단계의 Requirements Analysis에서 UML Diagram을 사용하지 않고, 시스템엔지니어링에서 사용하는 체계적인 요구사항 도출 방법을 적용하여 기능/성능뿐만 아니라 Life-Cycle전체에 걸친 요구

사항을 도출하였다.

- 시스템엔지니어링을 “어떻게 수행할 것인가?”에 대한에 구체적인 방법을 제시하였다.
- Text 기반의 요구사항을 UML로의 자연스러운 개발이 되도록 하였다.

실시간 시스템의 Analysis 단계에 시스템 엔지니어링 프로세스를 적용함으로써 실시간시스템 전체를 고려하였다. 더불어 체계적인 요구사항 개발 프로세스로 인하여 시스템의 안전성과 신뢰성이 확보된다.

시스템엔지니어링 기법을 적용함으로써 개발초기 시스템적인 사고로 시스템 전체를 고려할 수 있다. 따라서 실시간 시스템 개발 초기에 시스템 요구사항이 도출되고 나아가 더 복잡한 유비쿼터스와 같은 대형 복합시스템 개발에 적용할 수 있을 것이며, 국가 경쟁력을 높일 수 있는 기반을 형성할 수 있게 될 것이다.

참고문헌

- [1] James N. Martin, "Overview of the EIA 632 Standard: Processes for Engineering a System", Proceedings of the Eighth Annual International Symposium, INCOSE, p157-166, 1998.
- [2] Jim Cooling, "Software Engineering for Real-Time System", Addison-Wesley, 2002 Page 66~69.
- [3] Craig Larman, "Applying UML and Patterns Third Edition", Prentice Hall PTR, 2004.
- [4] Impact analysis and change management of UML models. O'Sullivan, Leeshawn Engineering, Electronics and Electrical. Carleton University (Canada), 2003.
- [5] Jim Cooling, "Software engineering for Real-Time Systems" , Addison-Wesley , 2003.
- [6] Muthiayen, Darmalingum Engineering, System Science. "Real-time reactive system development: A formal approach based on UML and PVS" ,Concordia University (Canada), 2000.
- [7] Dr. Douglass Guided Tour Through The Wonderland of Systems Engineering, UML and Rhapsody. I-Logix, 2002.
- [8] James N Martin, "Analysis of the Rational Unified Process", INCOSE, 2002.
- [9] 이중윤, "시스템 요구사항 정의 프로세스 및 도구 개발", 아주대학교 박사논문, 2004.