

A System Design Process Tailored for Reverse Engineering

Taehun Yoon

Department of Systems Engineering,
Ajou University, Woncheon-dong,
Yeontong-gu, Suwon-si 443-749, Korea
minus044@lycos.co.kr

YoungWon Park

Department of Systems Engineering,
Ajou University, Woncheon-dong,
Yeontong-gu, Suwon-si 443-749, Korea
ywpark@ajou.ac.kr

Copyright © 2009 by Taehun Yoon & YoungWon Park. Published and used by INCOSE with permission.

Abstract. This paper discusses a system design process using a reverse engineering. The Reverse Engineering approach, if possible is a cost-effective and easy approach to use in a system design. All industries use this approach consciously or unconsciously to reduce the system development risk. It can be a part of formal process, simple requirements reuse, or adoption of industry standard. The reverse engineering approach can be considered as an effective system design method in immature systems engineering environments. This paper proposes a system design process using reverse engineering which can be tailored for large complex system development projects. The proposed process composed of two stages to product system specification generation. The reverse engineering stage is performed to define functional and physical architecture of legacy system used as reference model when they are not available. The reengineering stage takes outputs of the reverse engineering stage to define the rest of logical and physical solutions.

초록. 이 페이퍼는 역공학을 통한 시스템 설계 프로세스에 대해 논의한다. 역공학적 접근은 시스템 설계에 있어 비용효과적이고 적용이 쉬운 방법이다. 우리는 의식적으로 또는 무의식적으로 개발의 위험부담을 감소하기 위해 역공학적 접근을 항상 사용하고 있다. 이러한 접근은 공식적인 프로세스의 일환일 수 있고, 단순한 요구사항의 재활용 또는 표준의 적용으로 나타난다. 성숙되지 못한 시스템 엔지니어링 환경에서 역공학적 방식은 효과적인 시스템 설계 방안으로 고려될 수 있다. 이 페이퍼는 대형 복합 시스템 개발 프로젝트에서 사용된 역공학 방식의 시스템 설계 프로세스를 제시하고 있다. 이 프로세스는 시스템 규격서를 개발하기 위한 두가지 단계로 구성된다. 역공학 단계는 참조 모델로 사용하는 기존 시스템의 요구사항 또는 규격서가 가용하지 않을 경우 기능적, 물리적 아키텍처를 정의하기 위해 수행된다. 다음 단계인 재공학 단계는 앞에서 정의된 출력물을 이용해 새로운 시스템 해결방안을 정의한다.

Introduction

Objective & Scope. 본 연구의 목적은 사례를 통해 도출된 역공학 방식의 시스템 설계 프로세스를 제시하고, 사례에서 식별된 문제와 교훈을 통해 그 프로세스를 정련하여 제시하는데 있다. 역공학적인 접근은 이미 보편적으로 사용되는 방법으로 기존 시스템에 사용된 요구사항을 재사용하거나, 규격서의 분석을 통해 적용되고 있다. 특히 국내 시스템 엔지니어링 환경과 같이 IPPT의 지원이 부족한 상황에서 시스템 엔지니어가 대상 시스템을 이해하고, 설계 대안을 도출하는데 있어 매우 중요한 역할을 한다. 그럼에도 불구하고, 이에 대한 연구와 사례 개발이 부족한 것이 현실이다.

본 연구의 범위는 대상 사례의 연구범위와 동일하게 적용된다. 대상 사례는 자기부상열차 실용화 사업의 일환으로 2007부터 수행된 자기부상열차 시스템 요구사항 데이터베이스 구축 과제이다. 기 과제는 자기부상열차 개발과 관련된 요구사항을 전산지원도구를 통해 데이터베이스화하기 위해 수행되었다. 따라서, 자기부상열차 운용과 지원에 관련된 모든 시스템을 대상 시스템으로 하며, 요구사항 정의와 해결방안 정의를 수행하는 시스템 설계를 주요 시스템 엔지니어링 업무로 수행하였다.

Approach. 본 연구에서 역공학 방식의 시스템 설계 프로세스를 정의하기 위해 표준 분석과 사례 분석을 중심으로 수행한다. 사례에서 참조된 시스템 엔지니어링 표준을 분석하여 적용이 용이한 순방향 시스템 설계 프로세스를 분석한다. 순방향 시스템 설계 프로세스의 적용에 있어 발생할 수 있는 문제점을 도출함으로써, 역공학 방식의 시스템 설계가 필요한 근거를 제시한다. 사례에서 수행된 활동을 기본으로 하여 역공학 방식의 시스템 설계 프로세스의 기본 틀을 정의한다. 이렇게 정의된 프로세스를 정련하기 위해 사례를 통해 얻은 교훈과 식별된 문제점을 정의한다. 사례를 통해 정의한 교훈과 문제를 바탕으로 역공학 방식의 시스템 설계 프로세스를 정련하여 제시한다.

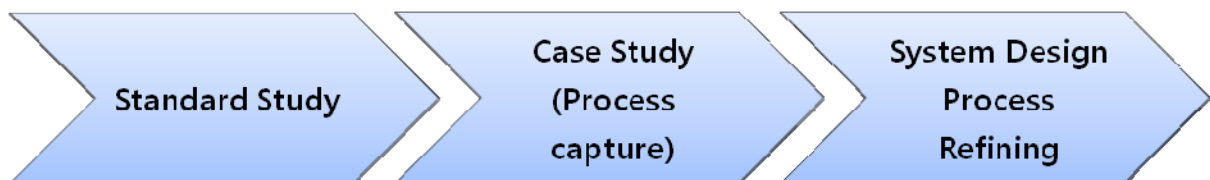


Figure 1 Approach

System Design Process Standard

Systems Engineering Standard. 역공학 프로세스의 정의에 앞서 순공학 프로세스를 정의할 필요가 있다. 시스템 엔지니어링 표준으로 EIA-632, ISO/IEC 15288, IEEE 1220 등이 존재한다. 이러한 표준들에 대한 비교분석은 기존의 연구자료를 통해 확인할 수 있다. 대표적으로 국제표준화기구의 합동기술위원회에서 발표한 자료에 따르면 표준이 다루는 상세도와 생애주기를 비교하였을 때, 개념화에서부터 운용까지 상위수준의 적용을 가장 잘 다루는 것이 EIA-632임을 알 수 있다. 따라서, 자기부상열차 시스템의 사례와 같은 대형 복합시스템의 시스템 수준 설계를 수행하기 위한 적용 표준으로 EIA-632를 선택하였다.

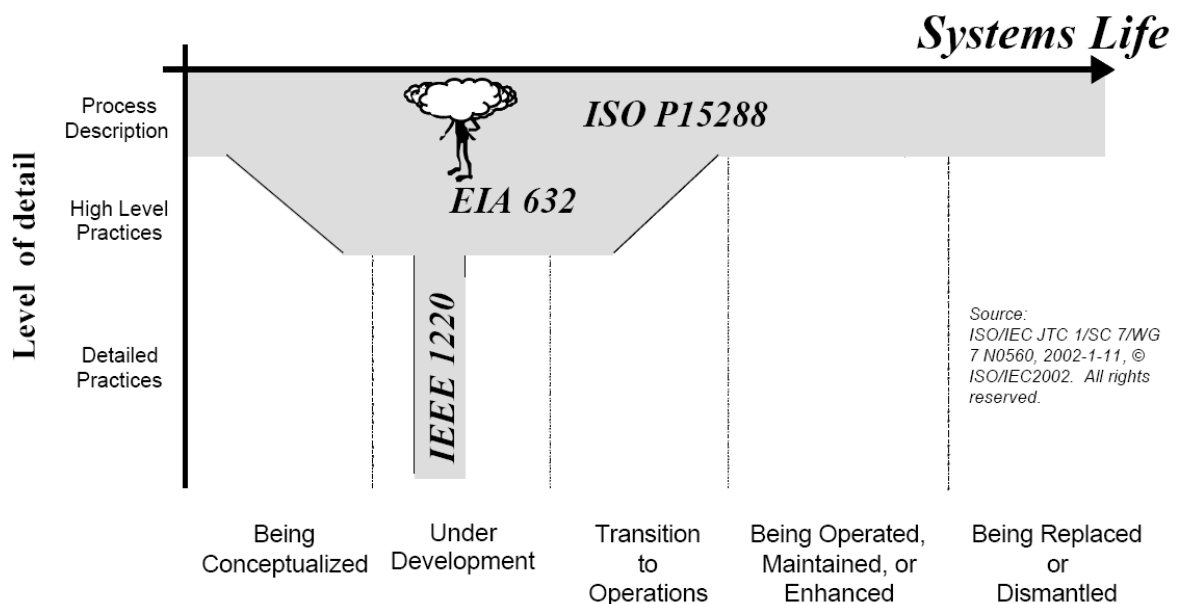


Figure 2 Scope of SE standards

대부분의 시스템 엔지니어링 프로세스는 what에 대한 기술로 이루어진다. EIA-632의 경우에도 마찬가지로, 시스템 엔지니어링에 필요한 과업들을 Requirement들로 정의하고, 각 Requirement들 간의 관계를 기술하고 있다. 이러한 기술 방식으로 인해 명확한 프로세스 식별이 다른 표준에 비해 어려운 것이 사실이다. 따라서 표준의 적용이 용이하도록 각 Requirement들의 관계를 도식화하여 정리하는 작업을 수행하였다.

사례를 기반으로 EIA-632의 시스템 설계 프로세스를 구성하는 각각의 Requirement들에서 대표 과업들의 유사성과 반복성에 따라 통합하고, 적용이 어려운 부분을 삭제하여, 시스템엔지니어링이 성숙되지 못한 분야에 적용 가능한 수준으로 단순화하였다.

그림 3에서 보이는 EIA-632의 시스템 설계 프로세스는 요구사항 정의 프로세스와 해결방안 정의 프로세스 두 단계로 구성되며, 요구사항 확인 프로세스와 시스템 분석 프로세스의 요구사항을 적용한다. 본 연구에서는 역공학에 초점을 두기 위해 시스템 분석 프로세스는 최소한으로 고려한다.

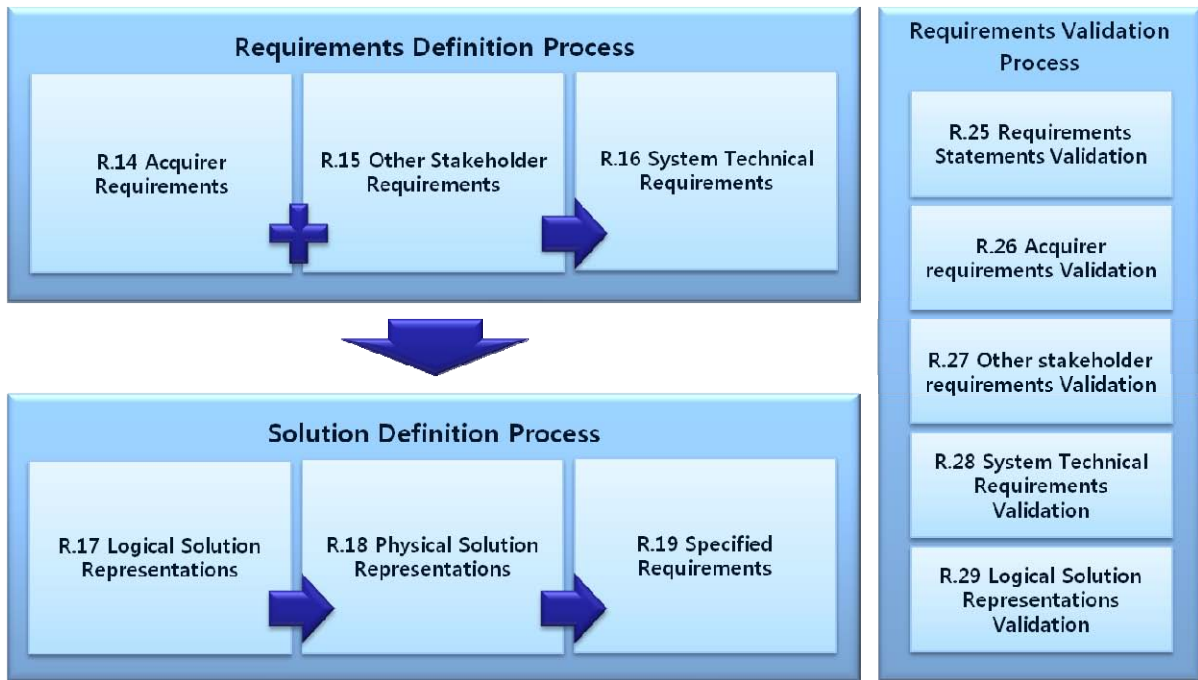


Figure 3 System Design Process

Requirements Definition Process. EIA-632의 요구사항 정의 프로세스는 requirement. 14 획득자 요구사항, requirement. 15 기타 이해관계자 요구사항, requirement. 16 시스템 기술 요구사항으로 구성된다. 각각의 requirement들은 요구사항 확인 프로세스의 수행을 필요로 한다. 여기에 속하는 것들이 requirement 25에서 28까지 이다. 그림 4에서 묘사하는 것과 같이, 요구사항 정의 프로세스의 요구사항들에서 반드시 필요한 대표 과업들을 분류하고, 요구사항 확인과 같은 중복적인 과업들을 통합하여 단순화한다. 이에 따라 요구사항 정의 프로세스를 정의하면 그림 4와 같이 묘사될 수 있다. 그림 5의 요구사항 정의 프로세스는 “모델기반 시스템 엔지니어링에 의한 체계 요구사항 분석 프로세스 연구”에서 보고된 내용을 바탕으로 수정한 것이다.

“모델기반 시스템 엔지니어링에 의한 체계 요구사항 분석 프로세스 연구”에서는 Requirement 26에서 28사이의 과업에서 Requirement 25에서 수행하는 두 대표 과업을 문장확인 과 집합확인으로 정의하여 적용하였고, 요구사항 정의 프로세스의 각

요구사항에 대한 확인 요구사항인 Requirement 26, 27 및 28을 계층화와 범주화로 정의하고 있다. 이러한 단순화는 각 확인 요구사항의 핵심 과업의 목적을 그대로 계승하고 있다.

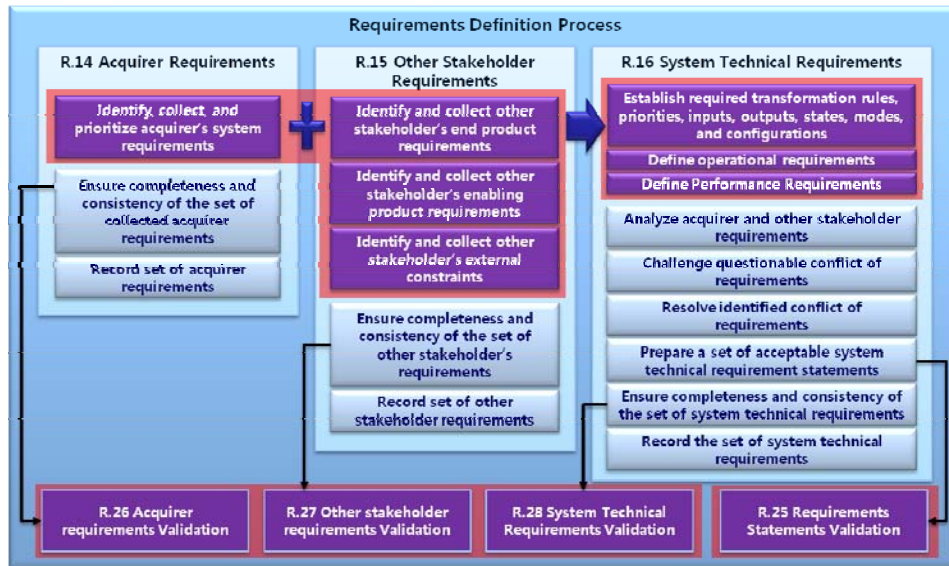


Figure 4 Refining Requirements Definition Process

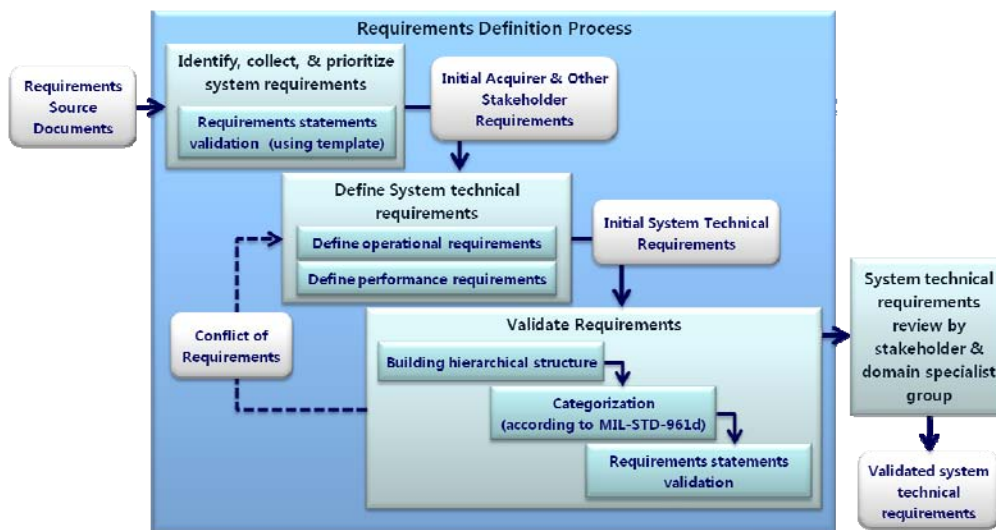


Figure 5 Simply Refined Requirements Definition Process

적용이 용이하도록 단순화된 그림 5의 요구사항 정의 프로세스는 획득자 및 이해관계자 요구사항 수집으로 시작된다. 요구사항의 수집에 있어 템플릿을 활용한 문장확인을 수행한다. 이 과업은 Requirement 25의 과업으로 요구사항의 기술에 있어 검토되어야 할 11가지 요소를 검토하는데 그 목적이 있다. 즉, 초기 요구사항의 수집과정에서부터

요구사항을 요건에 맞게 기술함으로써, 이후 확인과정에서 소모될 노력을 최소화할 수 있다. 이러한 작업은 요구사항 확인 통합 템플릿을 활용하여 용이하게 수행될 수 있다.

수집된 최초 이해관계자 요구사항은 기술적 용어를 사용한 기능적 및 비기능적 요구사항으로 작성되어 초기 시스템 기술 요구사항으로 정의된다. 정의된 초기 시스템 기술 요구사항들은 계층화, 범주화, 집합확인을 거친다. 계층화는 Requirement 26에서 28의 주요 과업으로 요구사항을 계층적으로 구조화하여 요구사항의 종적 추적성을 확보하는 과업이다. 범주화는 MIL-STD-961D와 같은 표준에서 제공하는 범주를 활용하여 요구사항이 누락 없이 작성되었음을 확인한다. 집합확인 은 요구사항 간의 중복과 상충을 확인하는 과업이다. 이러한 확인 과정을 거쳐 확인된 시스템 기술 요구사항을 정의한다.

확인된 시스템 기술 요구사항은 각 분야의 전문가들과 이해관계자들을 통해 시간적, 기술적, 비용적 타당성을 평가 받아 확인된 시스템 기술 요구사항으로 확정된다.

Solution Definition Process. 해결방안 정의 프로세스는 앞의 요구사항 정의 프로세스에서 도출된 시스템 기술 요구사항을 입력으로 규정된 요구사항을 도출하는 과정을 다룬다. 해결방안 정의 프로세스를 도출하는 과정도 앞의 요구사항 정의 프로세스와 같이, 표준에 기술된 대표 과업들을 분석 및 통합하여 실제 적용이 용이하도록 정련하였다.

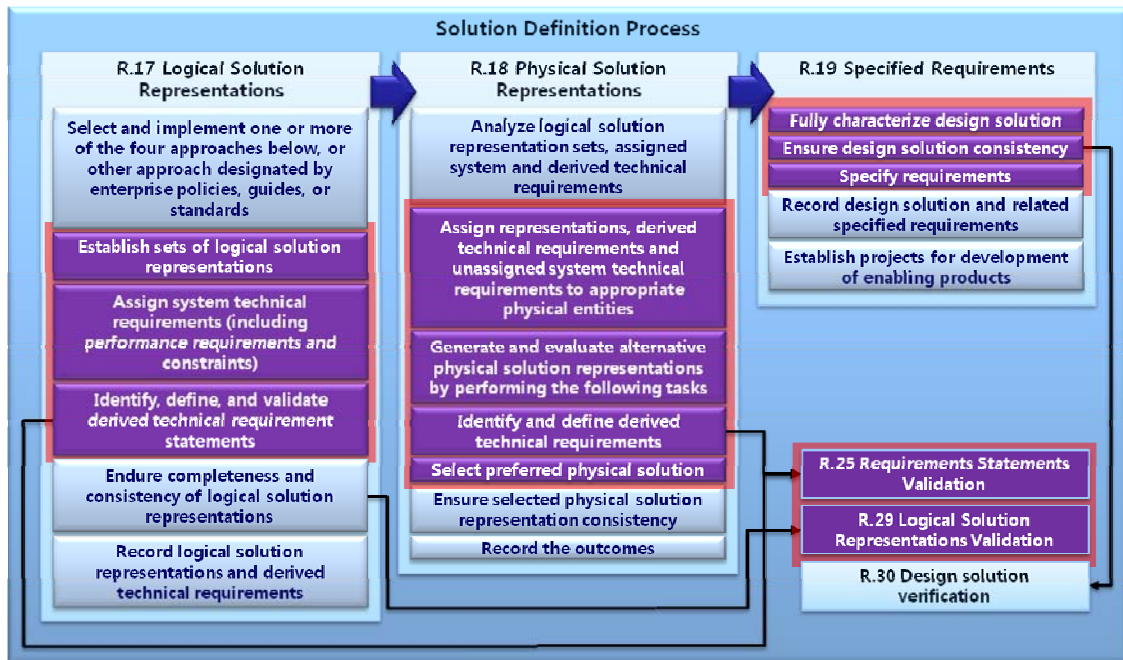


Figure 6 Refining Solution Definition Process

그림 7은 단순화하여 정련된 해결방안 정의 프로세스를 묘사하고 있다. 해결방안 정의 프로세스를 구성하는 논리적 해결방안은 시스템 기술 요구사항에 맞는 해결방안의 추상적인 정의를 위해서 기능 분석, 객체지향 분석, 구조적 분석, 정보 모델링 등의 방법이 사용된다. 이러한 방법으로 시스템 기술 요구사항을 논리적 해결방안의 구성요소인 하부 기능, 하부 기능 그룹, 객체, 그리고 데이터 구조 등으로 할당한다. 이 과정에서 시스템 기술 요구사항으로부터 나온 성능 요구사항과 제약사항 역시 할당된다. 시스템 기술 요구사항의 할당에서 상부 및 하부 추적성이 확인되지 않는 논리적 해결방안에 대해 유도된 기술 요구사항을 생성하여 다시 할당하거나 물리적 해결방안 프로세스로 전달한다. 논리적 해결방안 확인 과업은 앞에서 정의한 계층화, 범주화, 문장확인, 집합확인을 거쳐 확인된다.

이렇게 정의된 논리적 해결방안과 할당되지 않은 유도된 기술 요구사항은 물리적 해결방안 정의 프로세스에서 물리적 구성요소에 대한 개념적 객체로 할당된다. 개념적 객체를 정의하는 과정은 물리적 해결 방안 대안 개발과 병행되어, 논리적 해결방안의 산출물이 대안 객체에 할당되도록 한다. 논리적 해결방안의 할당을 통해 유도된 기술 요구사항을 도출하고 이를 시스템 기술 요구사항이나 논리적 해결방안에 추적시킨다. 도출된 물리적 해결방안은 계층화, 범주화, 문장확인, 집합확인을 거쳐 확인된다.

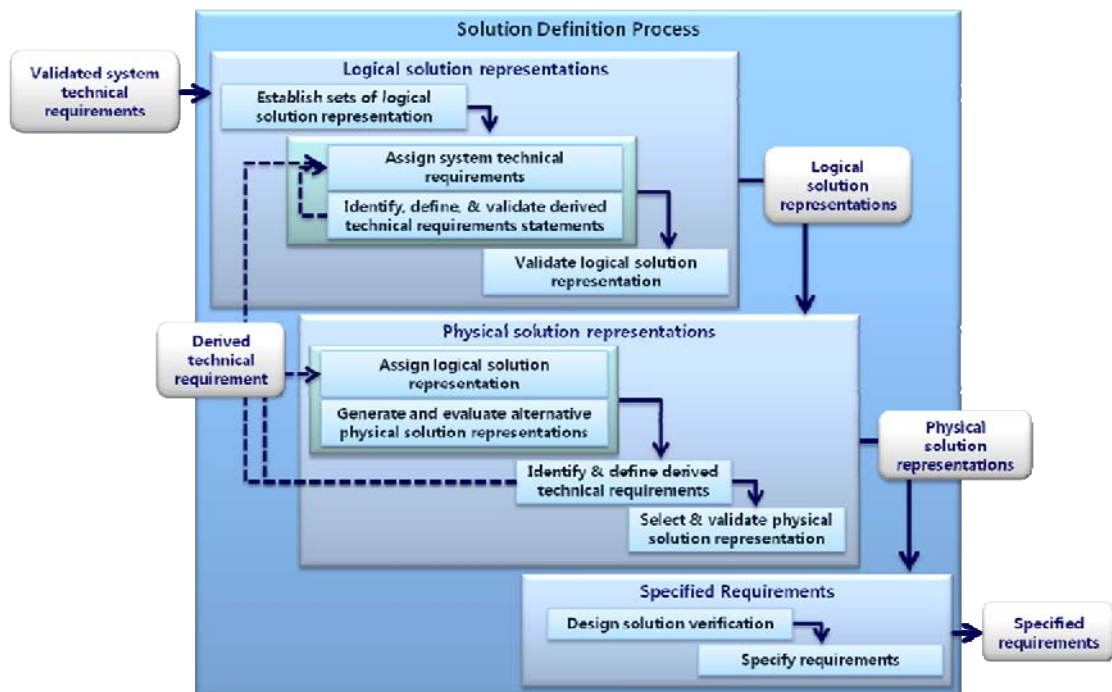


Figure 7 Simply Refined Solution Definition Process

정의된 논리적 해결방안과 물리적 해결방안에 따라 설계 해결방안이 도출되면, 육안검사, 시뮬레이션, 시연, 시험 등의 방법을 통해 이를 검증한다. 검증을 통해 해결방안과 요구사항간의 추적성을 확인하고 누락, 상충 등을 평가 및 수정한다. 검증을 거친 해결방안은 조직의 표준에 따라 규격서로 문서화 된다.

A Problem of System Design Process Application

Problems. 앞에서 제시한 시스템 설계 프로세스는 사례를 기반으로 정련한 최소한의 과업들로 구성되었다. 그러나, 순공학적 시스템 설계 프로세스만을 적용하는데 몇 가지 현실적인 문제가 따른다.

첫째, 시스템 엔지니어링이 적용되는 시점에 있어 표준 프로세스를 적용하는데 문제가 있다. 국내의 시스템 엔지니어링 적용 사업의 경우, 아직까지 시범사업의 성격이 강하다. 이로 인해 실질적인 시스템 엔지니어링 효과를 얻기보다 가시적인 성과만을 고려하여 사업을 계획하는 문제가 야기된다. 결국, 하부 구성품의 개발과 동시에 시스템 엔지니어링 활동이 수행되어, 시스템 엔지니어링의 산출물을 전체 시스템 개발에 적용하는데 한계가 발생하게 된다. 결국 이러한 경우에 각 구성품 개발에 사용된 규격서들을 정리하여 가공하는 비효율적인 활동으로 시스템 엔지니어링 활동이 수행되게 된다.

둘째, 시스템 엔지니어링 활동에 있어 중요한 역할을 수행하는 통합제품개발팀의 구성이 미비하고, 시스템 엔지니어링 팀의 구성이 다학제적 성격을 갖추지 못한다. 때문에 초기 요구사항 도출에 있어 대상 시스템의 전 생애와 각 분야에 걸친 완전한 요구사항의 개발이 힘들고, 도출되는 이슈에 대한 절충분석과 검토 기간이 필요이상의 지연되게 된다. 이러한 문제점들은 대부분 시스템 엔지니어링 환경에 의해 야기되는 문제들로 단기적으로 해결이 어려운 문제임이 분명하다.

셋째, 도구와 방법론에 대한 지원이 부족하다. 시스템엔지니어링을 적절히 지원하는 전산도구들이 많이 개발되어 있거나 개발 중에 있다. 그러나 실제 사업에서 이러한 전산지원 도구와 방법론들의 유용성에 대한 인식이 부족하여 지원이 제한적인 것이 현실이다.

PMTE 패러다임에 따르면, 환경, 도구 및 방법론이 적절히 프로세스를 지원하지 못한다면, 아무리 훌륭한 프로세스라 할지라도 올바르게 적용되기 힘들다. 앞에서 지적한 문제들은 결국 프로젝트의 성공 지표인 시간, 성능, 비용과 직접적으로 관련된 것들이다. 몇몇 시스템엔지니어링 선진국을 제외한 대부분의 국가들이 시스템엔지니어링 도입에 오랜

시간을 소비한다. 이것은 도입 초기에 시스템엔지니어링 적용의 유용성을 입증하기 어렵기 때문으로 생각할 수 있다. 결국, 시스템엔지니어링 환경을 개선하고 그 성숙도를 높이기 위해서는, 시간과 비용에 대한 제한된 지원 속에서도 올바른 결과를 도출할 수 있어야 한다.

Concept of Reverse Engineering. 역공학적 접근은 순공학적 접근이 갖는 위험부담을 줄이고, 시간과 자원 제약사항을 달성하기 위한 방안으로 소프트웨어뿐 아니라 기계, 전자분야에서도 널리 사용되는 방법론이다. 시스템 엔지니어링에 있어서도 앞에서 정의한 문제점들을 해결하기 위한 대안으로 역공학적 접근을 고려할 수 있다.

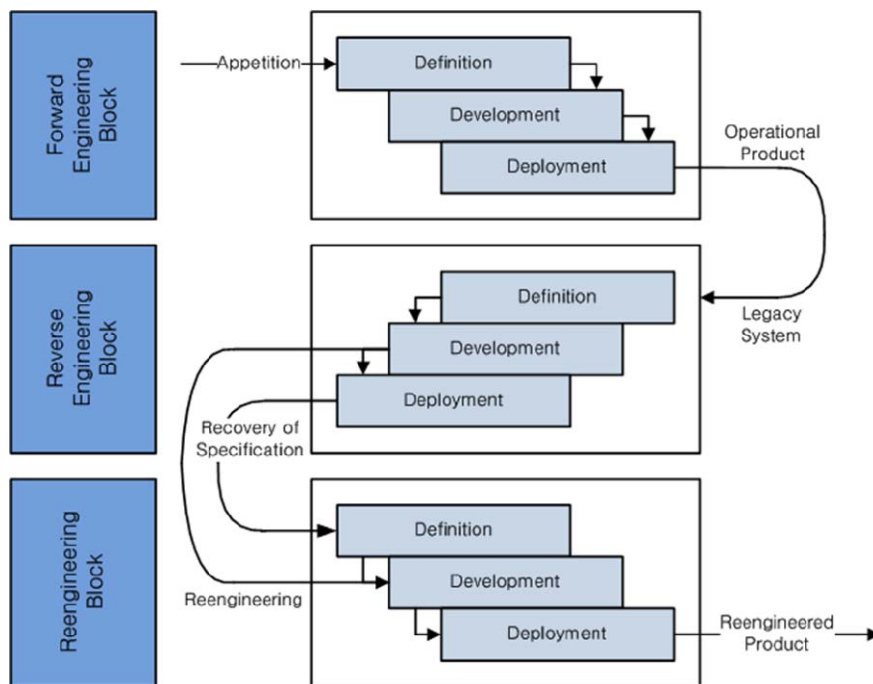


Figure 8 Concept of Reverse Engineering

역공학적 접근방법은 기존 시스템이 존재하는 경우 효과적인 개발 방법으로써 그 유용성은 Andrew P. Sage가 제안하였다. Sage가 제안한 기본 역공학과 재공학의 프로세스는 그림 4와 같다. 역공학 과정에서 달성해야 하는 중요한 목표의 하나는 기존 시스템 규격서에 대한 이해이다. 이를 기반으로 재공학 과정에서 새로운 시스템 요구사항을 달성할 수 있는 새로운 시스템 규격서를 개발하게 된다. 이러한 과정은 기존 시스템에 대한 벤치마킹 과정으로 볼 수 있다.

역공학에 의한 재공학의 개발 방법론이 만들어진 배경은 처음부터 순공학에 의한 개발보다 개발 기간, 자원, 성능 및 위험요소와 같은 프로젝트 4대 목표를 효과적으로 달성하기

위해서 이다. 또한 기술 환경의 변화 속도가 최근에 더욱 빠르게 변하며, 시간적, 자원적, 기술적 관점에서 개발에 대한 환경이 더욱 경쟁적으로 변화하고 있다. 이러한 개발환경에서 시스템 요구사항을 효과적으로 달성하기 위해서는 이와 같은 역공학과 재공학에 의한 시스템 개발 및 개선을 수행하는 것이 필요하다고 Sage는 제안하고 있다.

Reverse Engineering Application Case Study

Overview. 자기부상열차 시스템 사례에서 적용된 시스템 설계 프로세스를 도식화하여 나타내면 그림 9와 같이 묘사할 수 있다. 이 역공학 프로세스는 앞에서 정의한 순공학 프로세스에서 시스템 분석과 확인 과업을 제외한 나머지 과업을 기초로 정의하였으며, 실제 적용이 용이하도록 단순화 하였다. 전체 프로세스는 재공학 블록에 포함되는 요구사항 정의 프로세스와 해결방안 정의 프로세스, 그리고 역공학 블록으로 구성된다. 본 사례에서 적용된 역공학 방식은 정형화된 프로세스로의 적용보다 기존 시스템을 참고하여 재공학에 활용하는 성격으로 볼 수 있다. 즉 앞에서 정의한 시스템 설계 프로세스의 과업들을 중심으로 수행하고, 역공학 블록을 통해 도출된 기존 시스템의 구조적 분석 결과를 참고하는 형식이다.

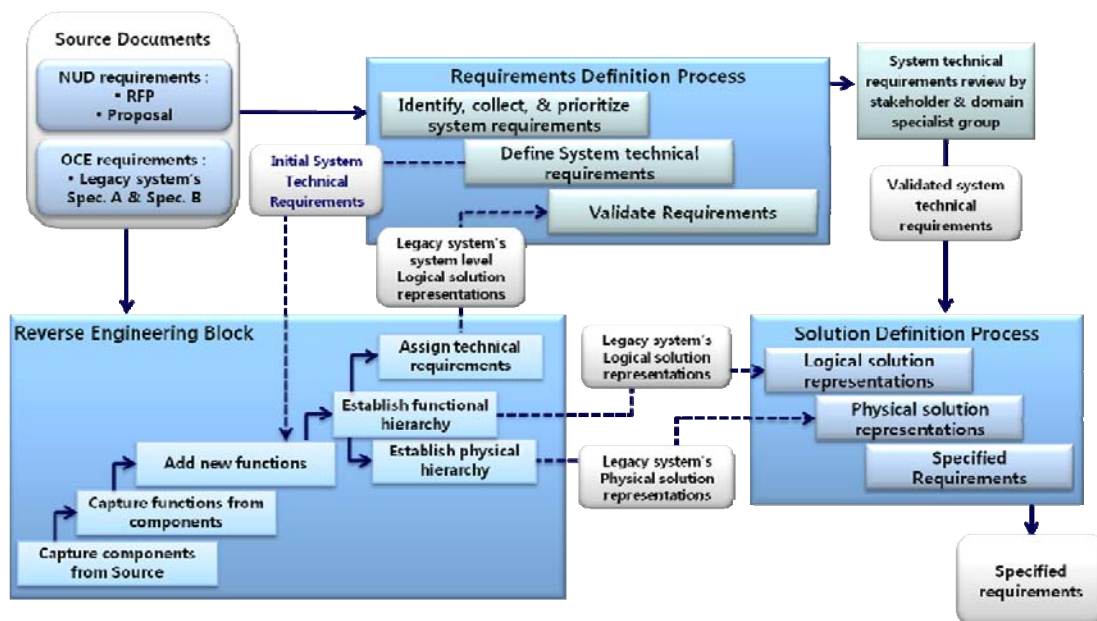


Figure 9 Reverse engineering application Process

이러한 역공학 활동은 대상시스템에 대한 이해를 높이고 관련된 요구사항을 직접적으로 재사용하는데 그 목적이 있다고 할 수 있다. 따라서 요구사항 정의 프로세스에서는 기존 시스템에 대한 요구사항과 새로운 시스템에 대한 요구사항을 모두 정의하고, 역공학

블록을 통해 정의된 기능 계층구조와 물리 계층구조를 참고하여 할당하는 방식으로 해결방안 정의 프로세스를 수행하게 된다. 이것은 해당 분야의 전문가가 시스템 엔지니어링 팀에 포함되지 않은 상황에서 대상시스템에 대해 이해를 높이는 것이 주요 목적이며, 짧은 기간에 대량의 요구사항을 도출하고, 브레인스토밍을 활성화 하는데 도움이 된다.

Reverse Engineering Block. 그림 10의 자기부상열차 시스템 사례의 역공학 블록은 기존 시스템의 규격서 분석을 통해 구성품들을 도출하고, 이를 기반으로 기능들을 식별함으로써 수행된다. 식별된 기능에 새로운 시스템에서 요구하는 기능들을 추가하여 새로운 기능 계층구조를 도출하여 낸다. 여기에 KJ법이나 DSM과 같은 방법이 적용될 수 있다. 초기 소스 문서들로부터 구성품의 계층구조를 먼저 도출하지 못하는 것은 시스템엔지니어링을 적용하지 않는 기존의 규격서들이 대부분 공간 중심의 물리계층구조를 갖고 기술되어 있기 때문이다. 따라서 기능계층구조를 도출한 이후에 이를 바탕으로 물리계층 구조를 도출한다. 여기서 도출된 기능계층구조는 논리적 해결방안으로 시스템 기술 요구사항의 계층화와 재공학의 논리적 해결방안 수립에 참조되고, 도출된 물리계층구조는 재공학의 물리적 해결방안 대안 생성에 참조된다.

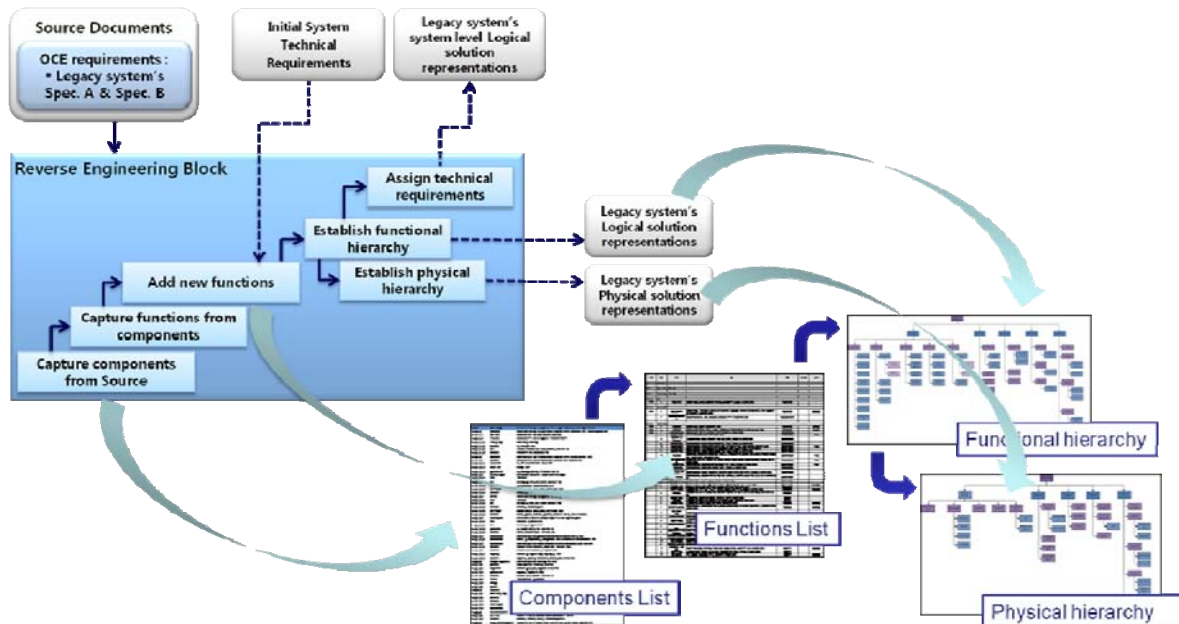


Figure 10. Reverse Engineering block

또한, 기능 및 물리 계층구조를 구성하는 각각의 요소들은 기존 시스템의 규격서로부터 도출한 기능 및 성능 요구사항과 설계 해결방안을 포함하므로 재공학에서 기술

요구사항으로 재활용할 수 있다.

대상 시스템은 자기부상열차 시스템과 그 운용 및 지원 시스템을 포함하는 대형 복합시스템(SoS)으로 차량 시스템, 신호 시스템, 통신 시스템, 전력 시스템, 선로구축물 시스템, 역사 시스템, 차량기지 시스템을 그 하부시스템으로 갖는다. 사례에서는 최상위 시스템 요구사항 88개를 비롯한 다음 5개 시스템에 대해 662개의 요구사항을 역공학을 통해 도출하였다.

Table 1 statistic of requirements captured by reverse engineering

system requirements	88
Vehicle subsystem requirements	251
Infrastructure subsystem requirements	142
Signal & communication subsystem requirements	112
Power subsystem requirements	157

그러나, 이 사례에서는 도구 측면의 지원이 부족하여 전산지원도구를 통해 시스템의 거동을 포함한 기능 아키텍처와 물리 아키텍처를 구축하고, 재공학에 활용하는데 제약이 존재했다. 따라서, 여기서 수행된 역공학의 목적은 기존 시스템에 대한 학습과 재활용 가능한 요구사항의 개발, 논리적 해결방안의 참조자료 생성, 물리적 해결방안의 대안 생성으로 정의할 수 있다.

Reengineering Process. 재공학 프로세스는 앞에서 정의한 요구사항 정의 프로세스와 해결방안 정의 프로세스를 따르며, 역공학 블록에서 정의된 기능 및 물리 계층구조를 바탕으로 수행된다.

역공학의 소스문서로 활용되는 기존 시스템의 Spec. A 또는 Spec. B는 대부분의 경우 1에서 5수준까지의 요구사항이 혼재되어 존재한다. 이러한 요구사항들을 역공학 블록에서 수립된 계층구조에 따라 할당된다. 그러나, 시스템 수준의 요구사항이 존재하지 않거나 새로운 시스템에 의해 변경되는 요구사항의 경우 Requirements 25의 요구사항 확인 요건에 맞추어 재공학 과정에서 다시 정의되어야 한다. 결과적으로 재공학 과정의 요구사항 정의 프로세스의 산출물인 시스템 기술 요구사항은 시스템 수준의 레벨1과 레벨2의 요구사항으로 작성된다.

해결방안 정의 프로세스는 역공학 블록에서 정의된 기능 및 물리 계층구조를 기초로

수행된다. 논리적 해결방안의 경우, 역공학 블록에서 정의학 기능계층구조를 기초로 논리적 해결방안이 수립되고, 시스템 기술 요구사항의 할당과정에서 새로운 시스템에 대한 기능요구사항뿐만 아니라 성능요구사항 및 제약사항이 할당된다. 물리적 해결방안의 경우, 역공학 블록에서 생성한 물리계층구조를 기반으로 물리적 해결방안 대안이 생성된다. 이 해결방안 정의 프로세스는 새로운 시스템에 대한 시스템 기술 요구사항이 적절히 할당될 때까지 반복적으로 수행된다.

System Design Process by Using Reverse Engineering

Lesson & Learned. 앞의 사례에서 사용된 역공학 방식의 시스템 설계 프로세스의 문제점과 교훈을 식별하여 더욱 효율적이고 효과적인 프로세스로 정련한다. 앞의 사례에서 식별된 역공학의 장점은 설계 해결방안의 개발에 있다. 이러한 장점을 더욱 살리기 위해서는 역공학에서 재공학에 이르는 모든 산출물이 전산지원도구를 통해 지속적 관리되어야 한다. 이것은 역공학에서 구축된 요구사항, 기능 및 물리 아키텍처가 재공학에서 그대로 재사용 가능하게 구축되어, 기존 시스템의 아키텍처에 대한 수정을 통해 새로운 시스템에 대한 아키텍처를 얻을 수 있어야 한다. 즉, 새로운 시스템 기술 요구사항을 바탕으로 이루어지는 설계 해결방안의 수정 및 아키텍처의 정련 작업이 더욱 효율적으로 수행되도록 역공학 산출물이 개발되어야 한다.

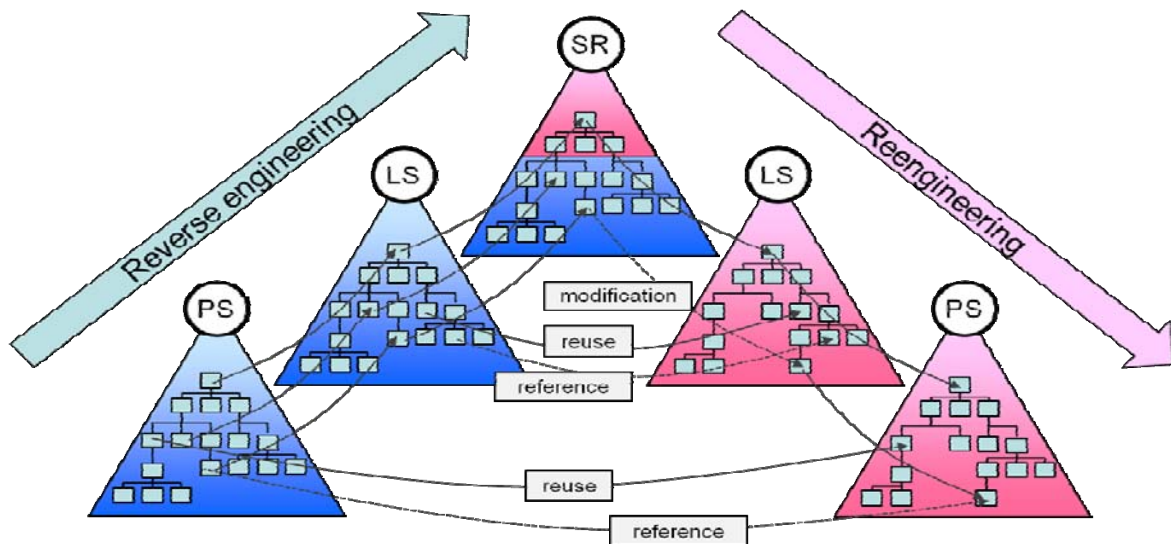


Figure 11 conceptual relation of architecture in reverse engineering and reengineering

역공학의 소스문서로 활용되는 대부분의 문서는 하부시스템에 대한 요구사항으로 구성되어 상위 시스템 수준의 요구사항을 도출하기 어렵다. 따라서, 역공학 과정에서

하부시스템 또는 구성품의 요구사항을 추상화하여 상위 수준의 요구사항을 생성할 필요가 있다. 또한, 재공학의 요구사항 정의 프로세스에서는 새로운 시스템에 대한 획득자 및 이해관계자 요구사항에 대한 순공학을 통해 요구사항을 개발함과 동시에, 유도된 요구사항을 개발하여 부족한 시스템 기술 요구사항을 개발하기 위해 노력해야 한다. 즉, 역공학 블록에서는 해결방안에 대한 참조모델 개발에 초점을 두어, 재공학에서 활용 가능한 더 많은 대안을 도출할 수 있도록 하는 것이 중요하고, 재공학에서 새로운 시스템에 대한 요구사항을 충분히 반영하기 위해 새로운 시스템에 대한 순공학적인 요구사항 정의 프로세스도 병행하여야 한다.

Conclusions

본 페이퍼에서는 성숙되지 못한 시스템엔지니어링 환경에서 수행되는 시스템 설계가 올바른 시스템 설계 해결방안을 도출하기 위한 방안으로 역공학적인 접근법을 제시하였다. 시스템 설계 프로세스에서 역공학의 목적은 기존 시스템에 대한 학습, 재활용 가능한 요구사항의 개발, 논리적 해결방안의 참조자료 생성, 물리적 해결방안의 대안 생성으로 정의할 수 있다. 사례를 통하여 이러한 목적들이 재공학 및 순공학 프로세스와 어떠한 관계를 갖는지 설명하였다. 또한 사례에서 식별된 교훈과 문제점을 바탕으로 역공학에서 전산지원도구를 통한 아키텍처의 활용과 재공학으로의 연계를 강조하였다. 시스템엔지니어링에 대한 환경적 지원과 IPPT의 구성이 어려운 시스템엔지니어링 도입 초기 또는 정착기에 시스템엔지니어링의 유용성을 보여, 향후 시스템엔지니어링 성숙도의 향상에 기여하기 위한 과도기적 접근방안으로 적용될 것으로 기대한다. 또한, 본 연구에서 고려되지 못한 부분으로 역공학 방식의 시스템 설계 프로세스에서 시스템 분석을 더욱 효율적으로 수행할 수 있는 방안에 대한 연구가 필요하다.

References

- EIA-632-1998, Processes for Engineering a System, electronic Industries Alliance, January 1999.
- James N. Martin, Systems Engineering Guide Book – a process for developing systems and products.
- Sheard, S. A. 1996. Twelve systems engineering roles. In Proceedings of the Sixth Annual International Symposium of the International Council on Systems Engineering (Boston, MA). Seattle: INCOSE.
- A. Terry bahill, , the systems Engineering started in the middle process : a consensus of systems engineers and project managers.
- Joong-yoon Lee, 2003, Automatic train control system development through application of reverse and re systems engineering process, ICASE 9-10.

INCOSE 2006 systems engineering handbook - a guide for system life cycle processes and activities, Version 3. ed. Cecilia Haskins.

INCOSE 2000 systems engineering handbook - a guide for system life cycle processes and activities, Version 2. ed. Cecilia Haskins.

국방과학연구소, 2005, 모델기반 시스템 엔지니어링에 의한 체계 요구사항 분석 프로세스 연구, 요구사항 정의 프로세스 3단계, p113

A System Design Process Tailored for Reverse Engineering

Taehun Yoon
Department of Systems Engineering,
Ajou University, Woncheon-dong,
Yeontong-gu, Suwon-si 443-749, Korea
minus044@lycos.co.kr

YoungWon Park
Department of Systems Engineering,
Ajou University, Woncheon-dong,
Yeontong-gu, Suwon-si 443-749, Korea
ywpark@ajou.ac.kr

Copyright © 2009 by Taehun Yoon & YoungWon Park. Published and used by INCOSE with permission.

Abstract. This paper discusses a system design process using a reverse engineering. The reverse engineering approach is a cost-effective and easy to use in a system design. Most of industries use this approach consciously or unconsciously to reduce the risk during system development. It can be a part of formal process, simple requirements for reuse or adoption of industry standard. The reverse engineering approach can be considered as an effective system design method in an immature environment of systems engineering. This paper proposes a system design process using reverse engineering which can be tailored for large and complex system development projects. The proposed process consists of two stages to generate system specification. The reverse engineering stage is performed to define the functional and physical architecture of legacy system which is used as reference model when the requirements of existing system or descriptions are not available. The reengineering stage takes outputs of the reverse engineering stage to define the rest of logical and physical solutions.

Introduction

Objective & Scope. The purpose of this study is to propose a system design process using reverse engineering approach. The proposed design process is refined through the problems and lessons from a case study. The reverse engineering approach is a common method for requirements reuse or specification analysis. The case of Korea where does not support IPPT, a system engineer plays an important role to understanding target system and carrying out alternative plans. In spite of this situation, there are not many researches and case studies.

The scope of this study is same as case study. The case study is about development of system requirements' database for maglev train by computer-aided design tool. Therefore, this case study considers all of systems that are related to maglev train operation and support. The main job of systems engineering in this case study is to design the system that performs requirements definition and solution definition.

Approach. In this study, we focus on standard analysis and case study in order to define the system design process by reverse engineering approach. We first analyze system engineering standards that are referred in case study, and then analyze forward system design process which is easy to apply. By recognizing problems that would have happen when applying to forward system design process, we explain the necessity of reverse engineering system design approach. Additionally, we define a framework of reverse engineering system design process based on performed activities in the case. We clarify the lessons and problems studied from case study to refine this process. Finally, an enhanced reverse engineering system design process is proposed.

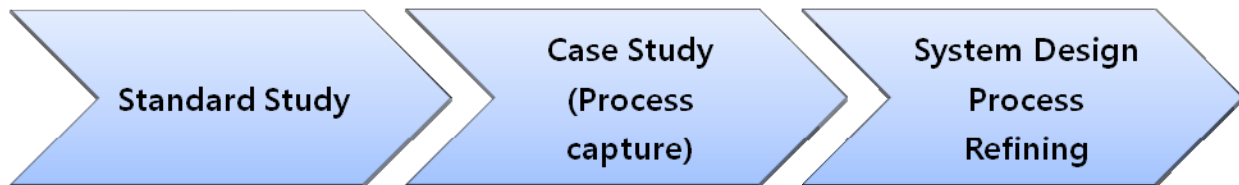


Figure 1 Approach

System Design Process Standard

Systems Engineering Standard. Before defining the reverse engineering process, it is required to define a forward engineering process first. There are system engineering standards such as EIA-632, ISO/IEC 15288, IEEE 1220 and so on. Comparison analyses among these standards have been studied in many literatures. Referring to joint technique committee of international standardization organization, EIA-632 covers well from being conceptualized until being operated in comparison of details and system life with other standards. Therefore, we choose EIA-632 standard for our case study, maglev train system which is a large and complex system design.

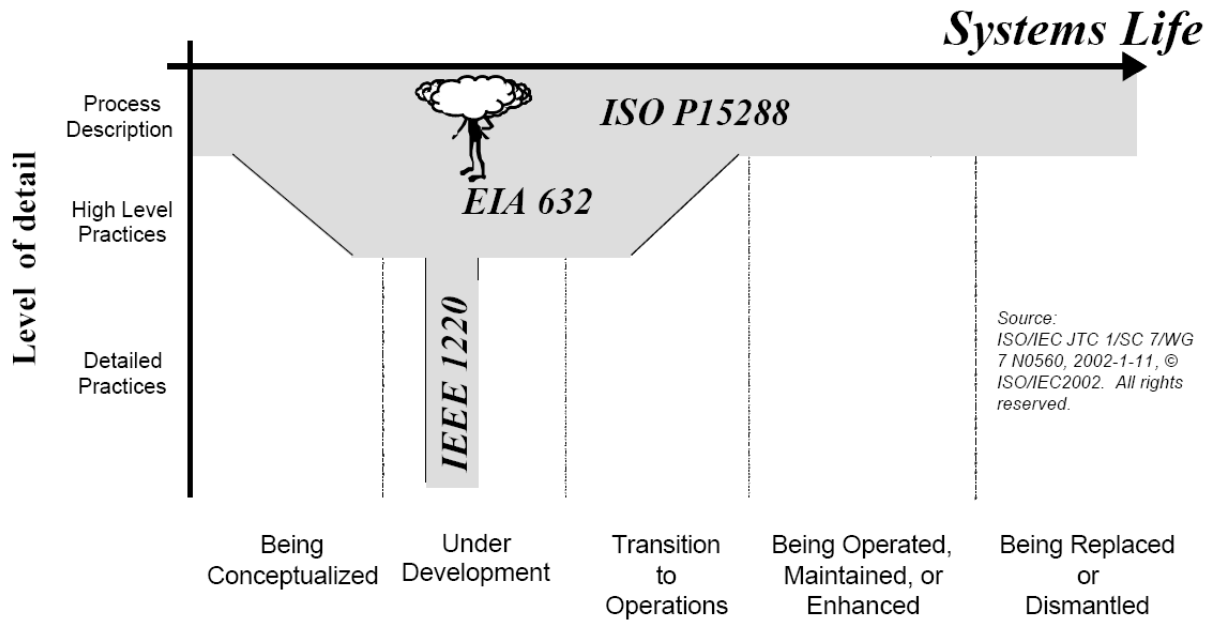


Figure 2 Scope of SE standards

Most of system engineering process is made by technique about 'What'. EIA-632 is also performed in a same sense that it first defines 'requirements' which is required assignments for systems engineering and then, describes the relationship among 'requirements'. Because of this descriptive way, it is difficult to identify the process precisely compared to other standards. Hence, we show the relationship among "requirements" in a diagram and summarize it with the purpose of applying to standards more conveniently.

Based on case study, each 'Requirement' in system design process for EIA-632 has been merged in terms of similarity and repetition of representative assignments, and deleted the part that is difficult to apply, so that the level of application is lowered to appropriate level for the field that is under the immature environment of systems engineering. As shown in <Figure 3>, the system design process of EIA-632 consists of 'requirements definition process' and 'solutions definition process'. It takes into account requirements of 'requirements validation process' and 'system analysis process'. As this study mainly focuses on reverse engineering, system analysis process is less considered.

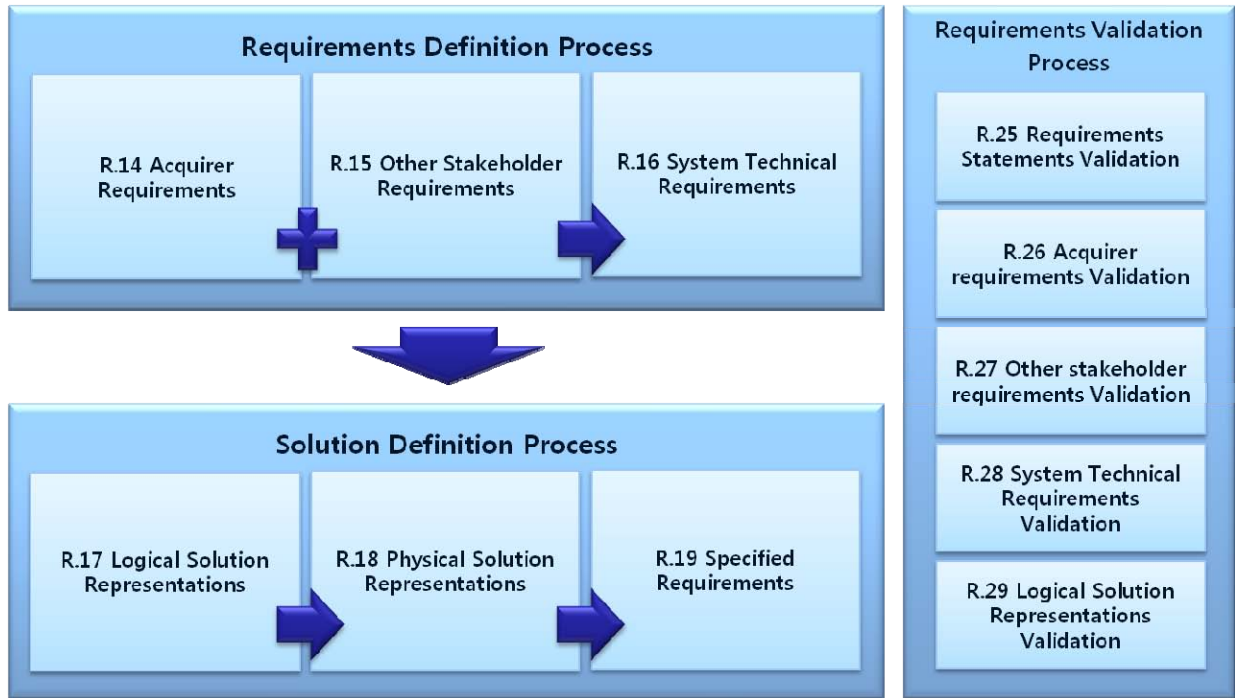


Figure 3 System Design Process

Requirements Definition Process. The requirements definition process of EIA-632 consists of 'Requirement.14 (acquirer requirements)', 'Requirement.15 (other stakeholder requirements)' and 'Requirement.16 (system technical requirements)'. Each requirement is needed to procedure 'requirements validation process'. It includes requirements from 25 till 28. As seen in <Figure 4>, it is first sorted compulsory assignments from requirements of 'requirements definition process' and simplified by merging repetitive assignments.

The requirements definition process described in <Figure 5> is modified from "A study on system requirements analysis process by model- based system engineering".

From this study, we define 'Requirement 25' as performance for two core assignments: statements validation and gathering validation. Requirement 26, 27 and 28 are validation requirements regarding each requirement of 'requirements definition process'. These are defined with hierarchy and category. This simplification still possesses the purpose of core assignments in each validation requirements.

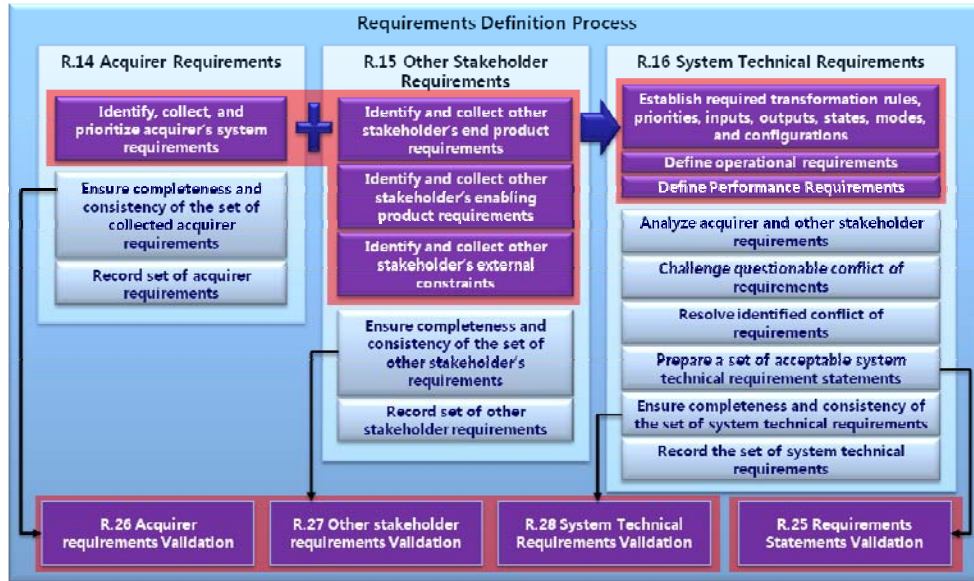


Figure 4 Refining Requirements Definition Process

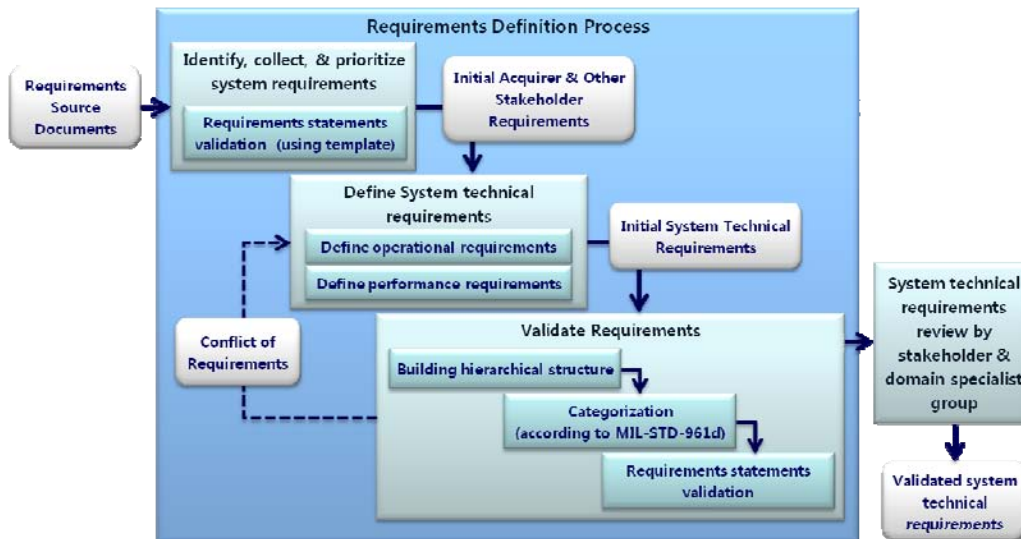


Figure 5 Simply Refined Requirements Definition Process

The simplified requirements definition process which has been seen in <Figure 5> starts from collecting requirements of acquirer and stakeholder. It executes 'statements validation' by templates application for collecting requirements. This assignment is 'Requirements 25'. It examines 11 elements for requirements statements. This means, as we describe requirements properly from initial collecting process, we can minimize the efforts for validation process. This work can be easily performed by integrated templates application of requirements validation.

The collected first stakeholder requirements that are divided into functional and non-functional requirements are defined as initial system technical requirements. The defined initial system technical requirements are examined with hierarchy, category and gathering validation. Requirement 26 and 28 are in charge of hierarchy process. They

form a hierarchical structure of requirements so as to secure the traceability of requirements. The categorization is for validating the case of requirements omission. Gathering validation is for validating repetition and contradiction among requirements. After bearing all these validation processes including review by stakeholder and domain specialist group, the validated system technical requirements would be defined.

Solution Definition Process. The solution definition process is the process for acquiring regulated requirements based on system technical requirements from previous requirements definition process. Deriving solution definition process is also implemented by analyzing and merging the core assignments from standards, and refining for practical application.

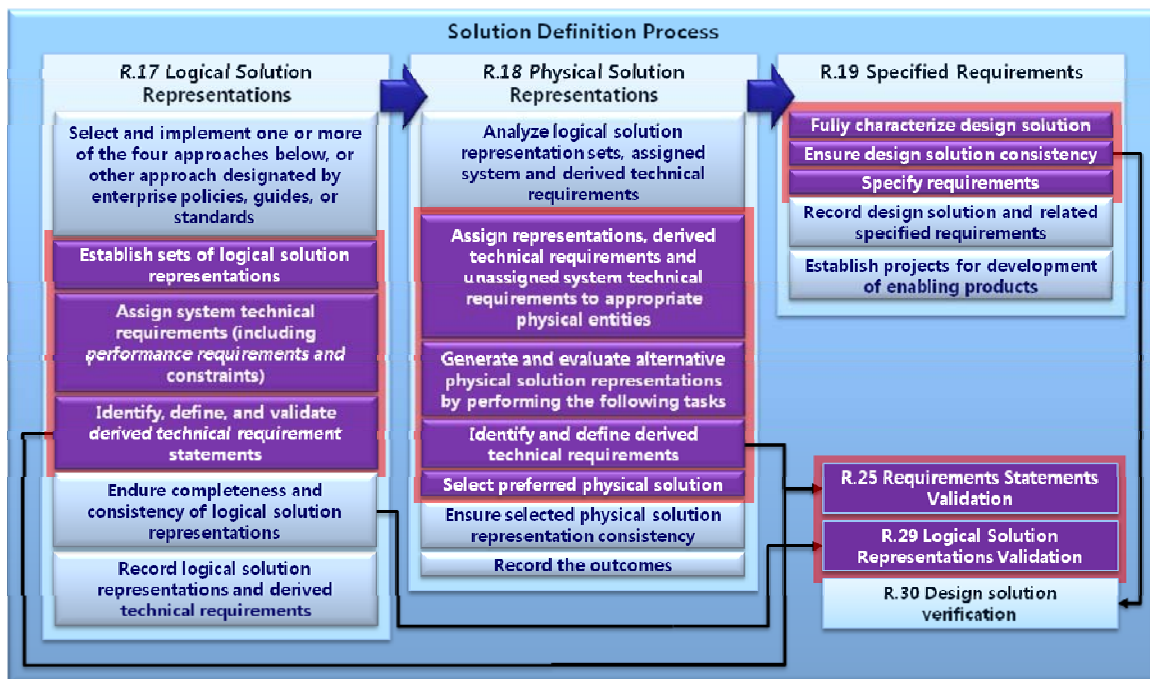


Figure 6 Refining Solution Definition Process

<Figure 7> shows simplified and refined solution definition process. The solution definition process consists of logical solutions and physical solutions. The logical solutions use functional analysis, object-oriented analysis, structural analysis and information modeling for defining abstract solutions. And system technical requirements are assigned to structural elements of logical solutions, sub-function, sub-function group, objects and data structure. In this process, the performance requirements and constraints derived from system technical requirements are assigned as well. At the process for assigning system functional requirements, it generates technical requirements and reassigns or delivers to physical solutions process for logical solutions that is not able to validate super or sub-track. The logical solutions validation is performed by hierarchy, categorization, statement validation and gathering validation.

Above defined logical solutions and non-assigned derived technical requirements are assigned as conceptual object regarding as physical structural elements in the physical solution definition process. A process for defining conceptual objects is implemented

simultaneously with development of physical solution alternatives. Hence, the output of logical solutions is assigned to alternative objects.

We can draw the technical requirements by assigning logical solutions and then trace back to system technical requirements or logical solutions. The result of physical solutions is verified by hierarchy, categorization, statements validation and gathering validation.

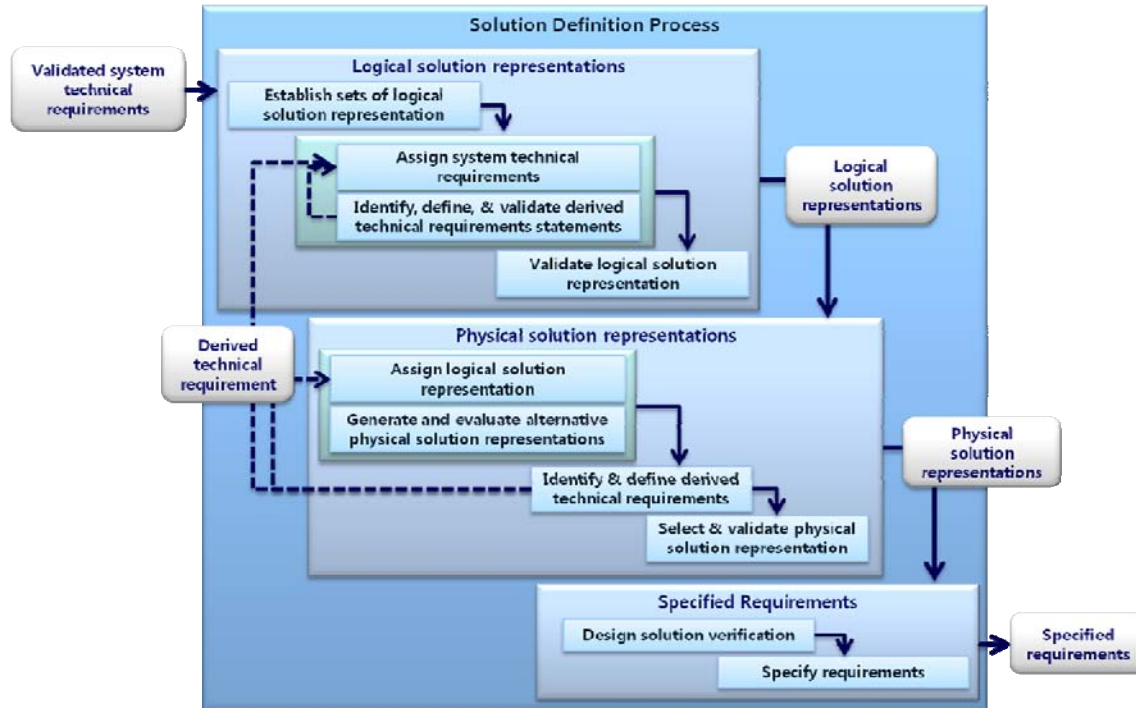


Figure 7 Simply Refined Solution Definition Process

When the design solution is achieved by defined logical solutions and physical solutions, it can be verified by simulation, demonstration, experiment and so on.

By verification, traceability between solutions and requirements can be investigated, and omission and contradiction can be evaluated and revised. The validated solution is committed to writing in terms of standard description of organization.

A Problem of System Design Process Application

Problems. The proposed system design process consists of minimal assignments that are refined from the case study. However, there are limitations for applying to forward engineering system design process.

First of all, a point of time that using system engineering has a problem for apply to standard process. Systems engineering application business in Korea is still just a model business. Therefore, it causes such problem that the business plan only considers visible outcomes rather than achieving actual system engineering effects. In this case, the system engineering activity is implemented with inefficient activity, suchlike arranging and processing the description that was used for developing each component.

Secondly, the structure of integrated product development team that plays an important role in the system engineering activity is defective, and the structure of system engineering team cannot have multidisciplinary characteristics. Therefore, it is difficult to develop the perfect requirements that consider the target system's whole life cycle and each specific part. Furthermore, it takes longer time to analyze and investigate issues of outcome. It is hard to solve in short time period since these problems are mostly caused by systems engineering environment.

Thirdly, the supports for tool and method are insufficient. Many of adequate computer-aided design tool to support systems engineering have been or being developed. However, the support is limited since the recognition and realization of usefulness of these computer-aided design tool and methods is insufficient. According to PMTE paradigm, if the environment, tool and method cannot support process properly, it is hard to apply in a right way, even if the process is excellent. These problems are directly connected to time, efficiency and cost those are an indicator of success in project.

Most of countries, excluding few of developed countries in system engineering, are spending long time to introduce system engineering. Perhaps it is because of difficulties of proving usefulness of system engineering application at an initial stage. In the end, it has to be able to achieve the right results in order to improve the systems engineering environment and grow the maturity, although the support for time and cost is limited.

Concept of Reverse Engineering. The reverse engineering approach is often used for reducing the risk that forward engineering approach has and satisfying time and resource constraints. Thus, it is common method for using in not only software industry but also machinery and electronic field. We can take into account the reverse engineering approach in order to tackle the problems of systems engineering.

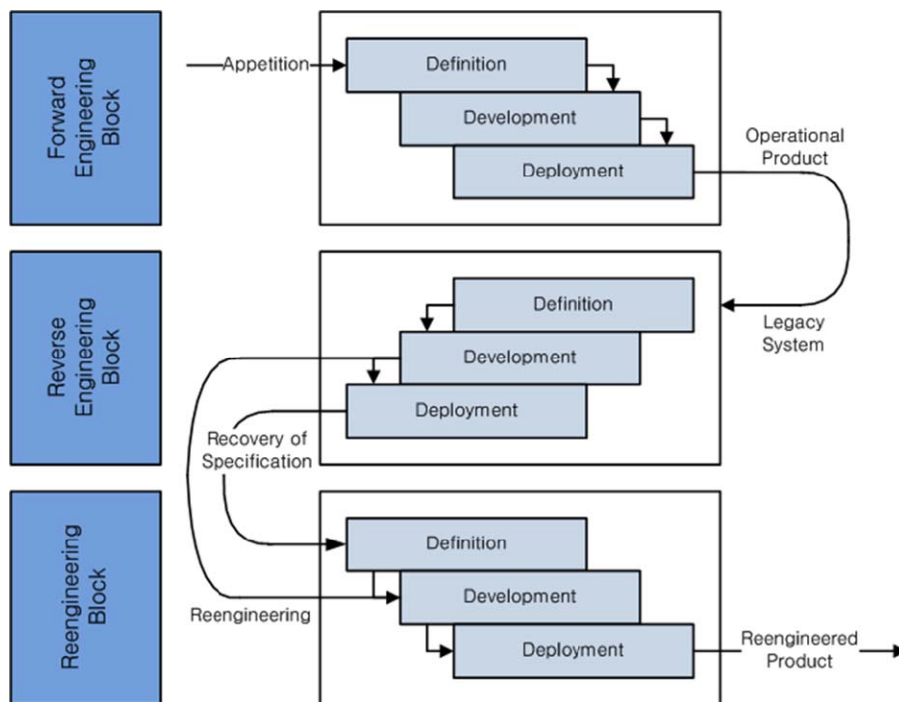


Figure 8 Concept of Reverse Engineering

The reverse engineering approach is effective when there is existing system. Its usefulness was proposed by Andrew P.sage. Basic reverse engineering and reengineering process that Sage proposed is shown in <Figure 4>. One of important goals in reverse engineering process is 'understand of existing system standard'. Based on this, it can develop the new system standards in the reengineering process that is able to satisfy the new system requirements. This is kind of a benchmarking process of existing system.

The background of reengineering development method by reverse engineering is to effectively accomplish the major 4 goals of development period, resources, efficiency and risk elements better than forward engineering. Moreover, as the technical environment is rapidly changing, development environment such as time, resource and technology is getting competitive. Sage has mentioned that it is unavoidable to develop and improve the system by reverse engineering and reengineering in order to effectively achieve the system requirements under the competitive development environment.

Reverse Engineering Application Case Study

Overview. The system design process that was adapted for maglev train system case can be described as shown in <Figure 9>. Total process consists of 'requirement definition process' which is included to reengineering block, 'solution definition process' and 'reverse engineering block'. The reverse engineering approach that is used in case study is closer to reengineering application. It mainly performs based on assignments of system design process that is defined previously, and refers the structural analysis result from existing system.

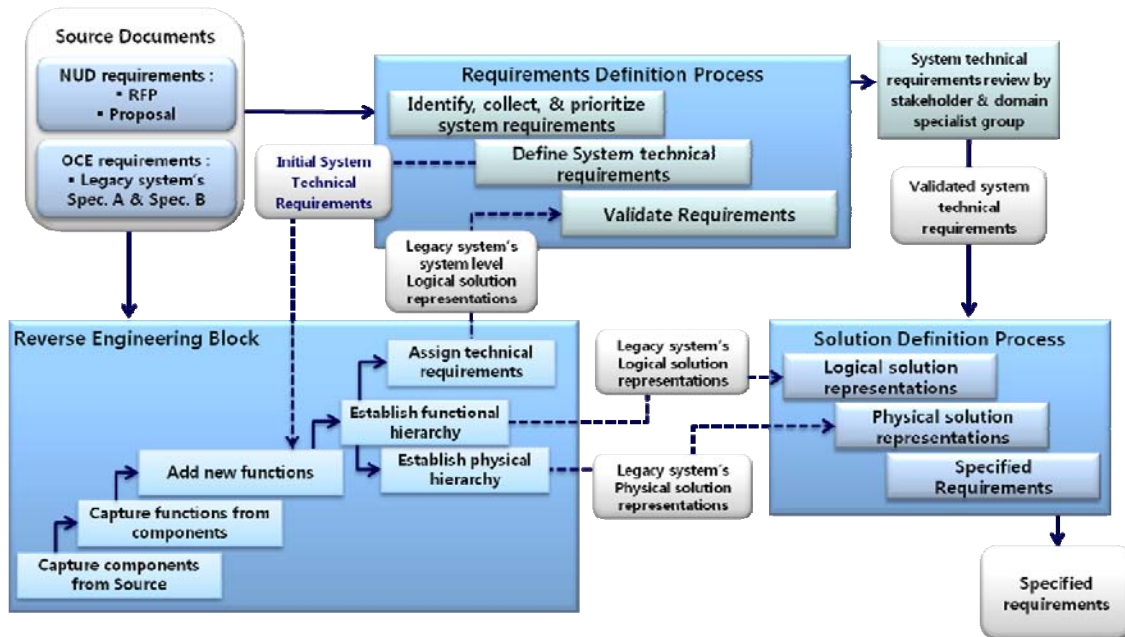


Figure 9 Reverse engineering application Process

The purpose of reverse engineering activities is to understand better about target system and reuse related requirements. Hence, both requirements for existing system and new system are defined at the requirement definition process, and then it performs solution definition process assigned by referring functional and physical hierarchical structure through reverse engineering block.

This aims to boost understanding of target system under the situation of when the domain specialist is absent in systems engineering team. In addition, it can carry out large amount of requirements within a short time period and is helpful for activating brainstorming.

Reverse Engineering Block. As shown in <Figure 10>, the reverse engineering block from the case study of maglev train carried out components by analysis of existing system description. Based on this, it can distinguish the functions.

Finally, those functions that are required from new system are added into distinguished functions. By this, hierarchical structure with new functions would be figured out. KJ or DSM method can be used here too.

The reason of non-enabling carrying the hierarchical structure of components from initial source documents is because of existing description that does not adapt system engineering and that mostly has space-oriented physical hierarchical structure. Therefore, we first accomplish the functional hierarchical structure and then carry out the physical hierarchical structure.

Hereby, the derived functional hierarchical structure is referred to establishing hierarchy of system technical requirements and logical solution for reengineering. The derived

physical hierarchical structure is referred to generating physical solution alternatives for reengineering.

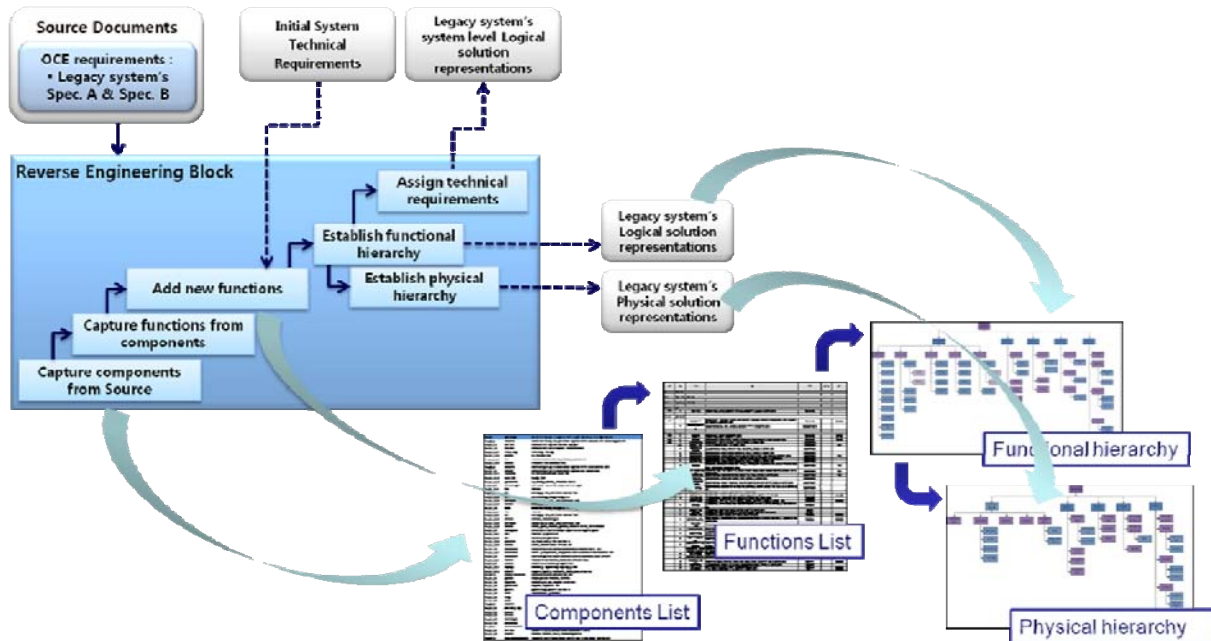


Figure 10. Reverse Engineering block

Furthermore, each element that composes a function and a physical hierarchical structure can be reused as requirements in reengineering since it contains function, performance requirements and design solutions from existing system description.

The target system is a large and complex system suchlike maglev train that has subsystems like vehicle system, signal system, communication system, power system, track infrastructure system, station building system, and vehicle station system. In the case study, including 88 requirements from uppermost system, total 662 requirements for 5 –system has been derived by reverse engineering approach.

Table 1 statistic of requirements captured by reverse engineering

system requirements	88
Vehicle subsystem requirements	251
Infrastructure subsystem requirements	142
Signal & communication subsystem requirements	112
Power subsystem requirements	157

However, in this case, there are constraints to use reverse engineering approach and we have to build a functional architecture and physical architecture by computer-aided design tool because support for tool is insufficient. Therefore, the goal of using reverse

engineering in this case study is to study existing system and develop the reusable requirements, generate references for logical solutions, and generate physical solution alternatives.

Reengineering Process. The reengineering process follows requirements definition process and solution definition process that were explained before, and performs based on functional and physical hierarchical structure defined in reverse engineering block. The Spec. A or Spec. B that is used as source documents in reverse engineering mostly exists with 1~5 various levels of requirements. These requirements are assigned to the hierarchical structure that is composed in reverse engineering block. However, when there are no requirements for system level or there are requirements that can be changed by new system, it has to be redefined according to requirements validation of Requirements 25. As a result, system technical requirements that are outcome of requirement definition process of reengineering process put in requirements of level 1 and level 2 which are system level.

The solution definition process is performed based on functional and physical hierarchical structure defined in reverse engineering block. In the case of logical solution, it is performed according to definition of functional hierarchical structure, while physical solution is generated physical solution alternatives by physical hierarchical structure. This solution definition process is performed repeatedly until system technical requirements are assigned appropriately.

System Design Process by Using Reverse Engineering

Lesson & Learned. In this chapter, we distinguish the problems and lessons that we learnt from system design process by reverse engineering, and refine the process more efficient.

An advantage of reverse engineering is development of design solution. It would be better managed all outcomes from reverse engineering and reengineering by computer-aided design tool in order to improve this advantage. The requirements and functional and physical architecture have to be reusable in reengineering when it is built in reverse engineering. It should be able to achieve the architecture for new system by revising architecture of existing system. Thus, output of reverse engineering has to be developed in order to perform more efficiently in revising design solution and refining architecture which is based on new system technical requirements.

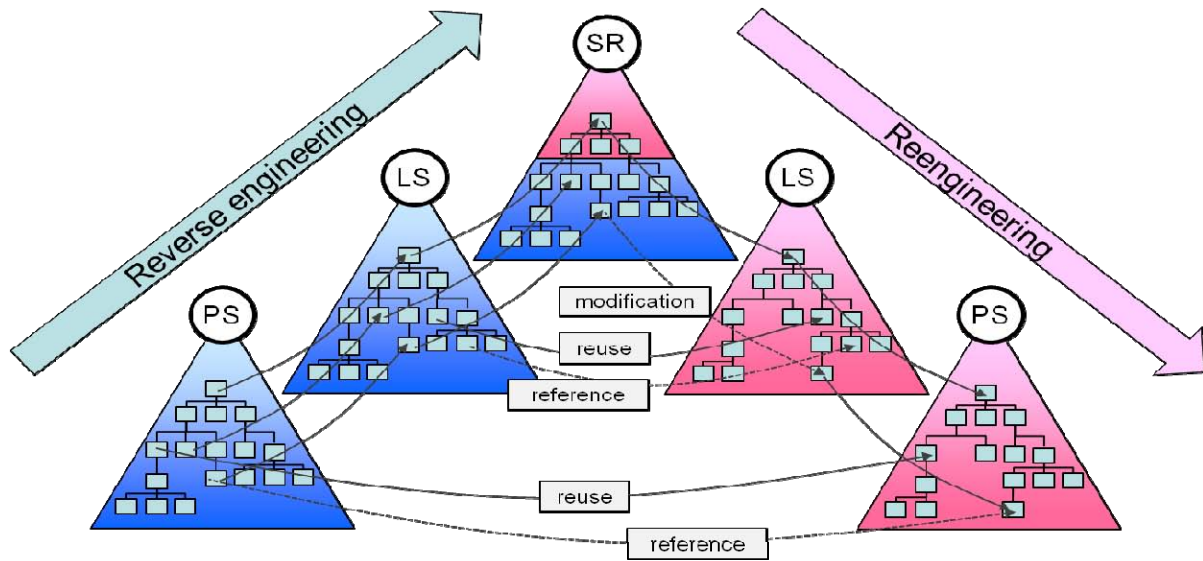


Figure 11 Conceptual relation of architecture in reverse engineering and reengineering

It is difficult to get the requirements for super system level since most documents that are used for reverse engineering source documents, since these are composed of requirements for subsystem. Therefore, in the reverse engineering process, it needs to generate high level requirements by abstracting subsystem or components requirements.

In the requirements definition process of reengineering, the requirements from system acquirer and stakeholder regarding as new system are developed by forward engineering, in the meantime it needs efforts to develop the system technical requirements by developing derived requirements. It is important to develop the reference model for solution in reverse engineering block since it can provide more alternatives that can be used in reengineering. To reflect sufficient requirements for new system in reengineering, it needs requirements definition process of forward engineering for new system.

Conclusions

This paper proposed the reverse engineering approach in order to carry out the solutions for system design that is performed under the immature environment of system engineering. The purposes of using reverse engineering in the system design process are to learn the existing system, develop the reusable requirements, generate reference data for logical solution and generate alternatives for physical solution.

We explained what kind of relationship exists between these purposes and reengineering and forward engineering process through the case study. In addition, we emphasized architecture application by computer-aided design tool and connectivity to reengineering in the reverse engineering based on lessons and problems from case study.

Since the results show that it is useful for environmental support of systems engineering and for initial or mature stage of systems engineering which is generally difficult to form the IPPT. It is expected to be a transition period approach to improvement of maturity of

systems engineering. Furthermore, as a future work, it is needed to improve efficiency of system analysis in the system design process of reverse engineering.

References

EIA-632-1998, Processes for Engineering a System, electronic Industries Alliance, January 1999.

James N. Martin, Systems Engineering Guide Book – a process for developing systems and products.

Sheard, S. A. 1996. Twelve systems engineering roles. In Proceedings of the Sixth Annual International Symposium of the International Council on Systems Engineering (Boston, MA). Seattle: INCOSE.

A. Terry Bahill, , the systems Engineering started in the middle process : a consensus of systems engineers and project managers.

Joong-yoon Lee, 2003, Automatic train control system development through application of reverse and re systems engineering process, ICASE 9-10.

INCOSE 2006 systems engineering handbook - a guide for system life cycle processes and activities, Version 3. ed. Cecilia Haskins.

INCOSE 2000 systems engineering handbook - a guide for system life cycle processes and activities, Version 2. ed. Cecilia Haskins.

Agency for defense development, Korea, 2005, A study on process of system requirements analysis by model-based system engineering, 3-step determine process for requirements, p 113.